Vers un cadre générique pour la recherche locale : application pour le Sudoku

ROADEF 2007, Grenoble, France

Tony LAMBERT, Eric MONFROY et Frédéric SAUBION

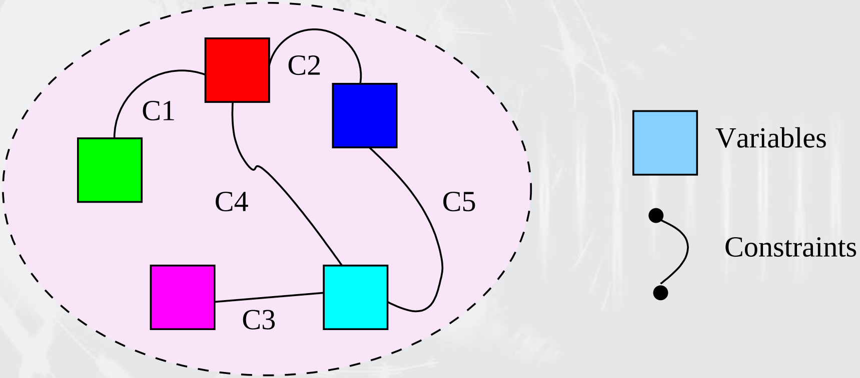
LINA, Université de Nantes, France

LERIA,Université d'Angers, France

Universidad Santa María, Valparaíso, Chile

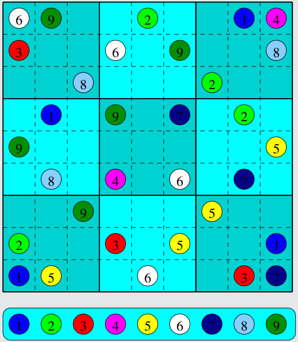
Mardi 20 février 2007

CSP (Constraint Satisfaction Problem)



Outline

- La recherche Locale pour la résolution de CSP
- Un cadre pour la Recherche Locale
- Algorithme Générique Itératif avec fonctions de réduction
- Résultats pour le problème du Sudoku
- Conclusion



Recherche Locale (RL) (1)

Définition :

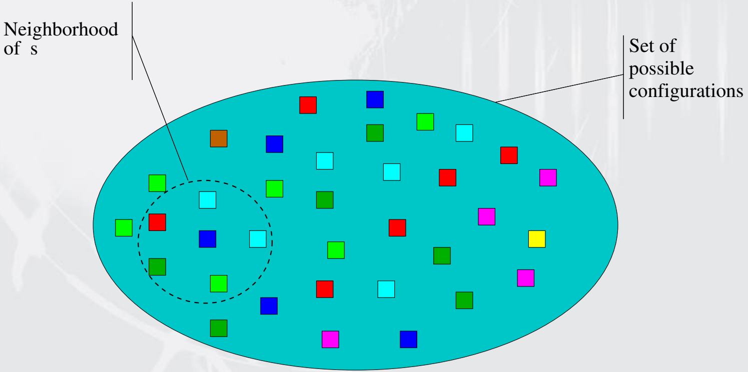
- Explore l'espace de recherche $D_1 \times D_2 \times \cdots \times D_n$
- Se déplace de voisins en voisins par rapport à une fonction d'évaluation
  - Intensification
  - Diversification

Propriétés :

- se concentrent sur des parties "prometteuses" de l'espace de recherche ;
- ne donnent pas de réponse face aux problèmes insatisfiables ;
- aucune garantie d'optimum global ;
- “rapides” pour trouver de “bonnes” solutions.

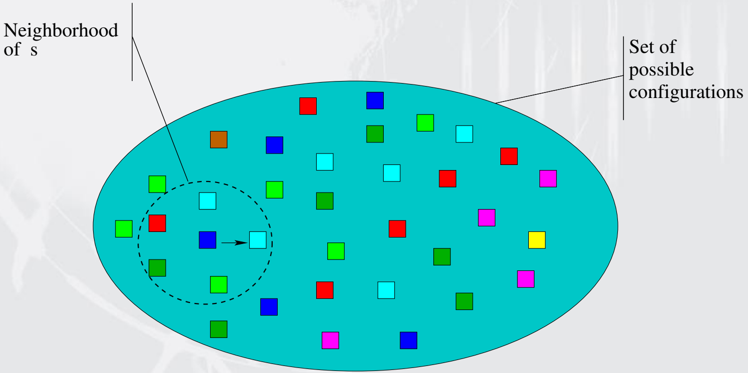
Recherche Locale (2)

- Espace de Recherche : l'ensemble des configurations possibles
- Outils : voisinage et fonction d'évaluation



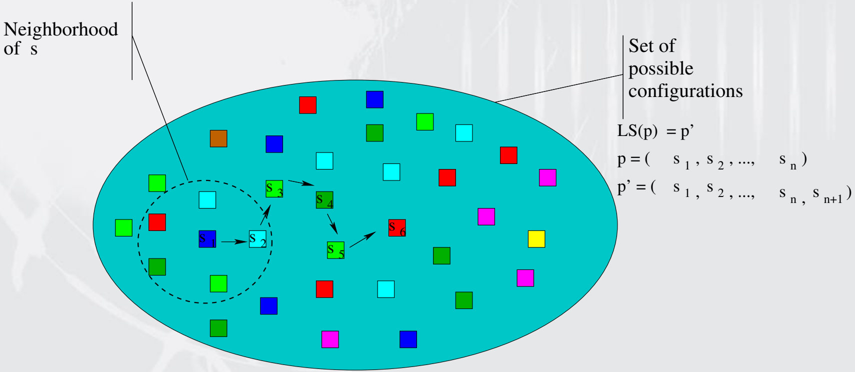
Recherche Locale (3)

- Espace de Recherche : l'ensemble des configurations possibles
- Outils : voisinage et fonction d'évaluation



Recherche Locale (4)

- Espace de Recherche : l'ensemble des configurations possibles
- Outils : voisinage et fonction d'évaluation



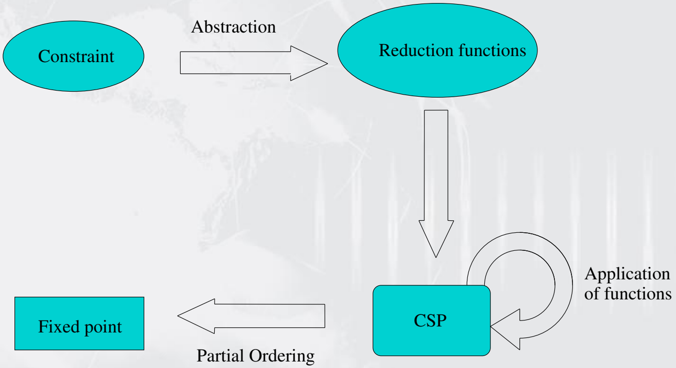
A cadre pour les techniques de Recherche Locale

- Idée :
  - Contrôle fin
  - Plus de stratégies
- Technique :
  - Décomposer les solveurs en fonctions de bases ;
  - Utiliser un algorithme générique pour la résolution hybride.

Notre but :

- Utiliser un modèle théorique existant pour la résolution de CSP
- Définir le processus de résolution

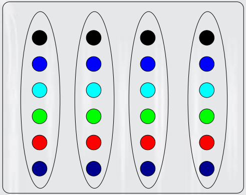
Modèle abstrait de K.R. Apt [CP99]



Un algorithme Générique

$F = \{$ ensemble de fonctions de réduction$\}$

$X =$ initial CSP
$G = F$
While $G \neq \emptyset$
choose $g \in G$
$G = G - \{g\}$
$G = G \cup update(G, g, X)$
$X = g(X)$
EndWhile

X = 

Le modèle théorique pour la résolution de CSP

Relation d'ordre pour la recherche locale

$$\mathcal{LS} =$$

$$\bigcup_{i>0} \{p = (s_1, \cdots, s_i) \in D^i \mid \forall j,\ 1 \leq j < i-1,\ s_{j+1} \in \mathcal{N}(s_j) \text{ et } s_1 \in D\}$$

Le modèle théorique pour la résolution de CSP

Relation d'ordre pour la recherche locale

Considérons une relation $\sqsubseteq_{ls}$ sur $\mathcal{LS}_D$ définie par :

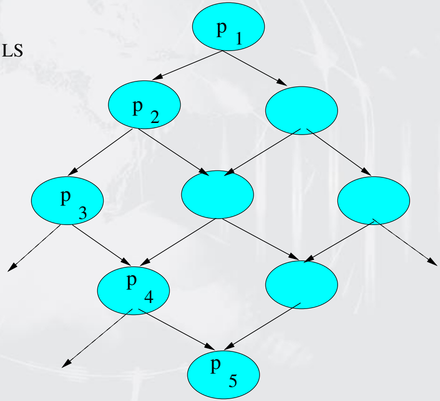
1. $(s_1, \ldots, s_n) \sqsubseteq_{ls} (s_1, \ldots, s_n)$
2. $(s'_1, \ldots, s'_m) \sqsubseteq_{ls} (s_1, \ldots, s_n)$ si $n > m$ et $\forall j, 1 \leq j \leq m, eval(s'_j) \neq 0$
   et $\forall i, 1 \leq i \leq n, eval(s_i) \neq 0$
3. $(s'_1, \ldots, s'_m) \sqsubseteq_{ls} (s_1, \ldots, s_n)$ si $eval(s_n) = 0, \forall i, 1 \leq i \leq n-1, eval(s_i) \neq 0$
   et $\forall j, 1 \leq j \leq m, eval(s'_j) \neq 0$

Le modèle théorique pour la résolution de CSP

Ordre Partiel :

Function to pass from one LS state to another

State of LS



Terminates with a fixed point : local minimum, maximum number of iteration

Ordre RL

Caractéristiques d'un chemin de RL

- notion de solution
- longueur maximale

Point de vue opérationnel : chemin = échantillons
Un échantillon :

- Dépend d'un CSP
- Correspond à un état de recherche locale

Voisinages et fonctions de mouvement

Voisinages :

- FullNeighbor : $V' = \{s \in D | s \notin V\}$
- TabuNeighbor : $V' = \{s \in D| \ \nexists k, n - l \leq k \leq n, s_k = s\}$
- DescentNeighbor : $p = (s_1, \ldots, s_n)$ and $V' = s \subset D$ s.t. $\nexists s' \in V$ s.t $eval(s') < eval(s_n)$

Mouvements :

- BestMove : $p' = p \oplus s'$ and $eval(s') = min_{s'' \in V} eval(s'')$
- ImproveMove : $p = p'' \oplus s_n$ and $p' = p \oplus s$ s.t. $eval(s') < eval(s_n)$
- RandomMove : $p' = p \oplus s'$ and $s' \in V$

Instantiation de l'algorithme GI

Nous pouvant maintenant introduire l'ensemble de fonctions F.

Tabu search :

- TabuNeighbor
- BestNeighBor

Random walk + Descent :

- FullNeighbor
- BestNeighBor
- RandomNeighbor
- DescentNeighbor
- ImproveNeighBor

Random walk :

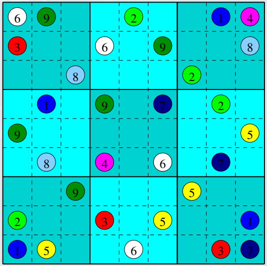
- FullNeighbor
- BestNeighBor
- RandomNeighbor

TabuSearch + Descent :

- TabuNeighbor
- DescentNeighbor
- ImproveNeighBor
- BestNeighBor

Le problème du Sudoku

- problème $n^2 \times n^2$
- n^4 variables
- contraintes *AllDiff*



	TabuSearch			RandomWalk		
n^2 x n^2	16x16	25x25	36x36	16x16	25x25	36x36
cpu time	3,14	115,08	3289,8	3,92	105,22	2495
deviations	1,28	52,3	1347,4	1,47	49,3	1099
mvts	405	3240	22333	443	2318	13975
	Descent + TabuSearch			Descent +RandomWalk		
n^2 x n^2	16x16	25x25	36x36	16x16	25x25	36x36
cpu time	2,34	111,81	2948	2,41	82,94	2455
deviations	1,42	55,04	1476	1,11	36,99	1092
mvts Avg	534	3666	20878	544	2581	14908

Conclusion et perspectives

- Un modèle générique pour la Recherche Locale
- Définir des stratégies

- Apprendre des stratégies dynamiquement
- Fournir plus d'outils dans un environnement générique

Vers un cadre générique pour la recherche locale : application pour le Sudoku

ROADEF 2007, Grenoble, France

Tony LAMBERT, Eric MONFROY et Frédéric SAUBION

LINA, Université de Nantes, France

LERIA,Université d'Angers, France

Universidad Santa María, Valparaíso, Chile

Mardi 20 février 2007