

# Modèles et algorithmes pour la reconfiguration de systèmes répartis utilisés en téléphonie cellulaire

Soutenance de thèse de doctorat

---

Renaud SIRDEY (Nortel, Heudiasyc)

UTC, 29 mars 2007

- 1998 : diplôme d'ingénieur de l'université de technologie de Compiègne en génie informatique et MSC en traitement du signal de l'université de Cranfield (G.-B.).
- 1998 à 2001 : concepteur logiciel sur le GSM/UMTS HLR/AuC (Nortel Maidenhead, G.-B.).
- 2001 à 2003 : responsable de l'équipe « tolérance aux pannes » pour la plateforme CNP du BSCe3 (Nortel Châteaufort).
- Depuis 2003 : architecte système BSC (Nortel Châteaufort).
- Mai 2004 : début de la convention Cifre (Nortel, UTC/Gradient, ANRT) dans laquelle s'inscrit le présent travail de thèse dirigé par MM. J. CARLIER et D. NACE (Heudiasyc), encadré par M. J.-O. BOUVIER (Nortel) et en collaboration avec M. H. KERIVIN (Limos).

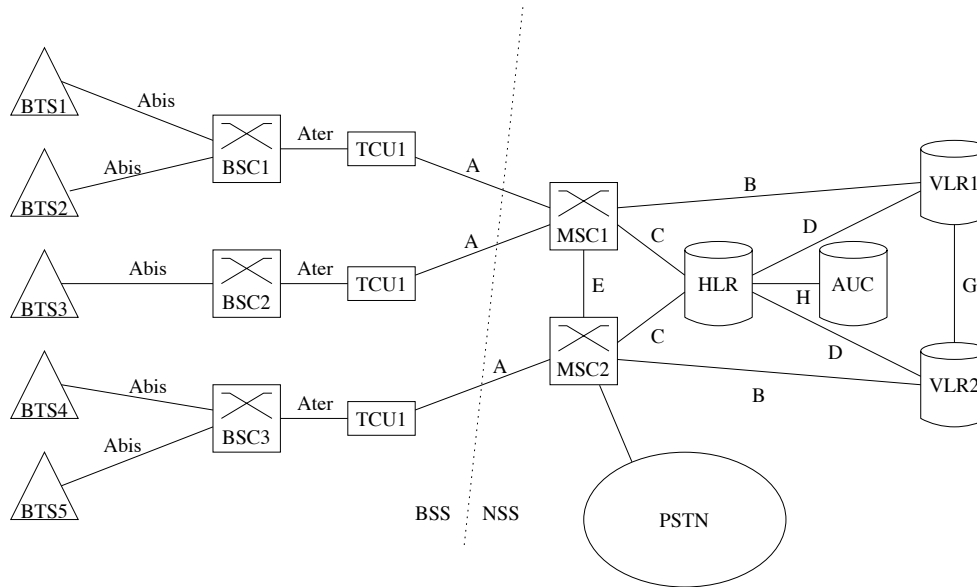
- Préliminaires.
  - Les réseaux GSM, le BSCe3.
  - Le problème de reconfiguration.
  - L'optimisation combinatoire.
- Étude du problème de reconfiguration.
  - Complexité, cas polynomiaux.
  - Méthode arborescente.
  - Recuit simulé.
  - Polyèdres et méthode de coupes et branchements.

# Préliminaires

- *Global System for Mobile communications.*
- 2 milliards d'utilisateurs dans plus de 200 pays.
- Dates clefs :
  - 1982 : création du Groupe Spécial Mobile.
  - 1990 : première version du standard.
  - 1991 : premier réseau commercial.
  - 1993 : premier million d'utilisateurs.
  - 1997 : GPRS.
  - 2000 : EDGE.
  - 2007 : Evolved EDGE.



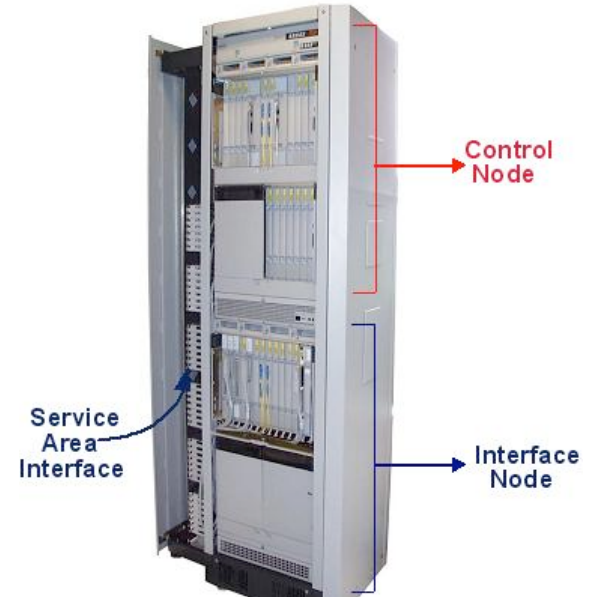
# Architecture simplifiée d'un réseau GSM 6/29



AuC : *Authentication Center* — BSC : *Base Station Controller* — BSS : *Base Station Subsystem*  
BTS : *Base Transceiver Station* — HLR : *Home Location Register* — MSC : *Mobile Switching Center*  
NSS : *Network & Switching Subsystem* — TCU : *TransCoding Unit* — VLR : *Visitor Location Register*.

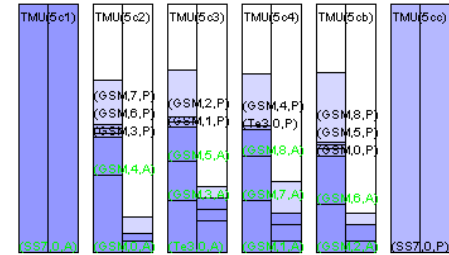
# Le BSCe3 : un autocommutateur réparti 7/29

- Nouvelle génération de BSC commercialisés par Nortel.
- Pilote jusqu'à 500 BTS.
- Écoule jusqu'à 3000 erlangs.
- Haute disponibilité :
  - Mécanismes sophistiqués de tolérance aux pannes.
- Flexibilité :
  - Le trafic est réparti entre des unités de traitement d'appels (TMU) que l'on peut ajouter ou enlever à chaud.

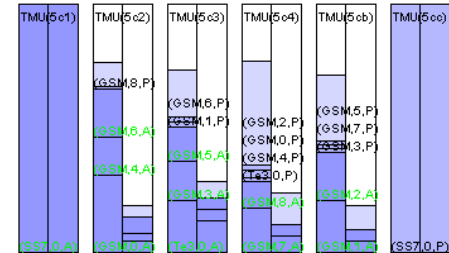


Aspect extérieur du BSCe3.

- Au démarrage, l'état du BSC possède de *bonnes* propriétés, notamment d'équirépartition de la charge [5].
- Ces propriétés se perdent au gré des défaillances des TMU car on ne s'autorise que l'utilisation de procédures simples et robustes pour traiter les défaillances.
- De plus, le BSC peut être démarré à vide.
- Il convient donc de pouvoir restaurer un état équivalent à l'état de démarrage du BSC sans le redémarrer.



État initial.



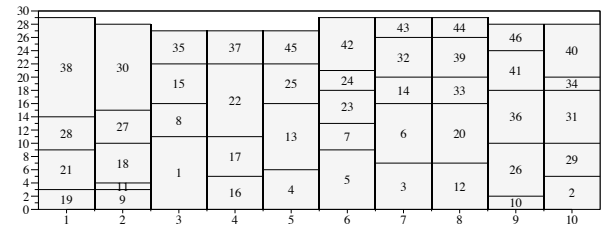
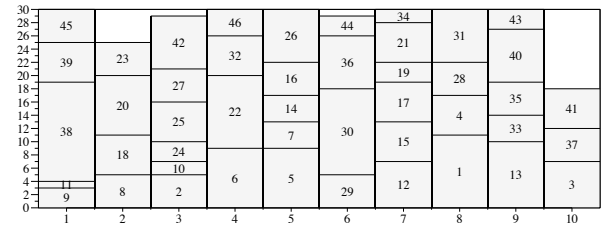
État après redémarrage de la TMU 0x5cb.



# Le problème de reconfiguration

9/29

- Soit un autocommutateur composé de processeurs sur lesquels tournent des processus.
- On cherche à faire passer le système d'un état à un autre, sans violation de capacité sur les processeurs.
- Un processus peut être :
  - Déplacé sans impact sur le service (*migration*).
  - Arrêté, puis redémarré (*interruption*).
  - L'interruption, à éviter, permet de résoudre les blocages.



État initial en haut, état final en bas.

- Un problème d'optimisation combinatoire consiste à trouver *un* meilleur élément dans un ensemble de taille *finie* mais astronomique.
- L'optimisation combinatoire est l'une des branches les plus appliquées des mathématiques.
- Dans un contexte où il convient d'utiliser efficacement des ressources interdépendantes (circuits, ports, liens, CPU, mémoire, etc.), ces problèmes sont omniprésents [1].



Le problème du voyageur de commerce (Applegate et al. — [www.tsp.gatech.edu](http://www.tsp.gatech.edu)).

# Les problèmes intrinsèquement difficiles 11/29

- On conjecture aujourd’hui qu’il existe des problèmes combinatoires que l’on ne peut *toujours* résoudre en un temps « raisonnable », ce sont les problèmes *NP-difficiles*.
- En présence d’un tel verrou, des compromis s’imposent.
- Deux faces d’une même pièce :
  - Résolution exacte : garantie d’optimalité sans garantie de temps.
  - Résolution approchée : garantie de temps sans garantie d’optimalité.



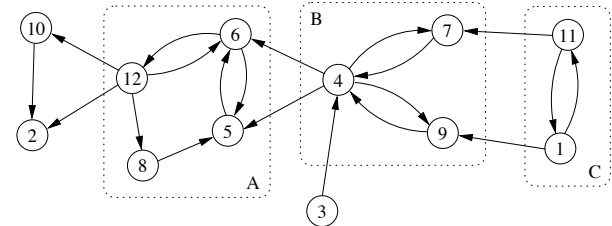
“I can’t find an efficient algorithm, but neither can all these famous people.”

Garey & Johnson, 1979.

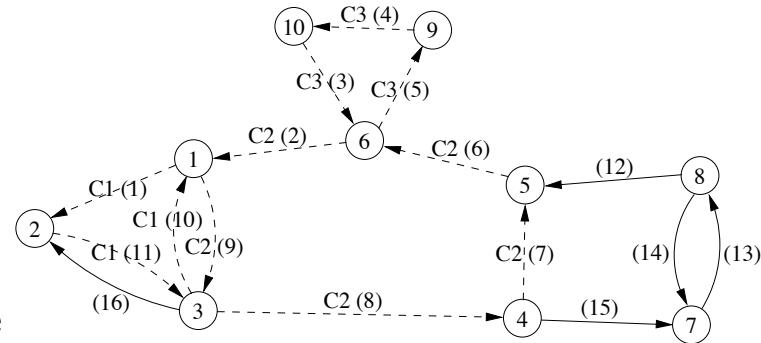
# Étude du problème de reconfiguration

# Complexité, cas polynomiaux [4]

- $NP$ -difficile au sens fort.
- Graphe de transfert,  $G = (V, A)$  :
  - $V$  : les processeurs.
  - $A$  : les déplacements.
- Cas polynomiaux :
  - Graphe de transfert sans circuit.
  - Cas homogène : une ressource, processus de consommations identiques.
  - Décomposition : les composantes fortement connexes de  $G$  peuvent être traitées indépendamment, dans l'ordre inverse d'une bonne numérotation.



Décomposition et forte connexité.



Cas homogène.



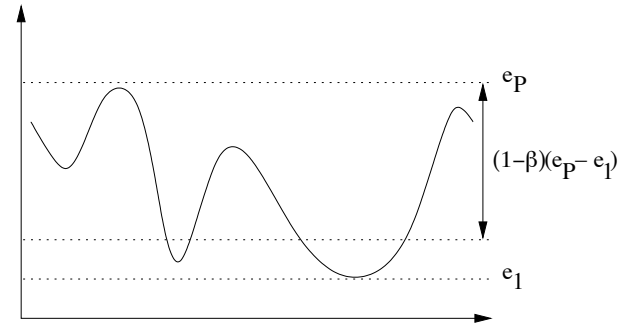
M	LB		Dom.		LB & dom.		
	#nodes	#nodes	#keys	#items	#nodes	#keys	#items
22	>18500000	>15900000	>316729	>606958	177542	6738	7905
21	16647308	>15500000	>224009	>454493	189618	7255	11178
16	12726	319552	8905	16232	2679	220	244
20	575391	>16100000	>210796	>510507	34829	2093	2573
17	1243750	488432	10821	22253	23635	1354	1968
19	265197	13217379	136421	480749	29891	1808	2162
18	14972721	5876920	66570	153435	55209	2685	4116
23	>23600000	>15000000	>334966	>627169	457337	18783	24298
20	1526411	>15700000	>215828	>481464	55045	2996	3611
17	143800	1609022	38846	86350	25814	1475	1971

- Une solution  $\omega$  d'un problème de minimisation est  $(\alpha, \beta)$ -acceptable si
$$P(c(\omega) \leq e_1 + \beta(e_P - e_1)) \geq \alpha.$$
- Soit  $e_1 \leq z \leq e_P$ , les solutions issues de la loi stationnaire de l'algorithme de Metropolis à la température

$$T_f(z) = \frac{\beta(e_P - z)}{\log N - \log(1 - \alpha)}$$

sont  $(\alpha, \beta)$ -acceptables.

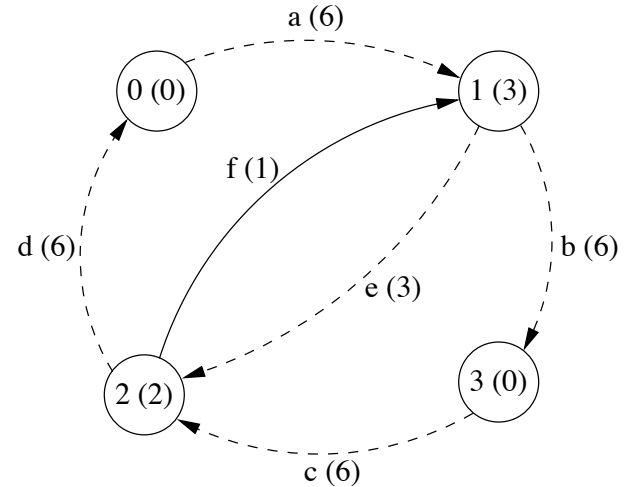
- Recuit simulé : convergence de proche en proche vers  $\pi^{(\infty)}(T_f(z^*))$ .



Principe de l'approximation différentielle.



- Fonction de voisinage : 2-OPT.
- $e_P = \sum_{m \in M} c_m$ , d'où
$$T_f(c^*) = \frac{\beta(\sum_{m \in M} c_m - c^*)}{\log |M|! - \log(1 - \alpha)}.$$
- Calcul de  $c(\omega)$  en  $O(|M|)$ .
- $|M|$  itérations par paliers.
- Loi de décroissance choisie telle que
$$|\pi_i^{(\infty)}(T_k) - \pi_i^{(\infty)}(T_{k+1})| \leq \delta.$$
- Nombre de paliers en
$$O(|M| \log |M| \sum_{m \in M} c_m).$$
- Complexité de l'algorithme :
$$O(|M|^3 \log |M| \sum_{m \in M} c_m).$$



Attention à ne pas « rater » l'optimum !

- Paramètres :  $\alpha = 0.95$ ,  $\beta = 0.05$ ,  
 $\delta = 0.1$ .

- Sur une base de 1020 instances  
« difficiles » :

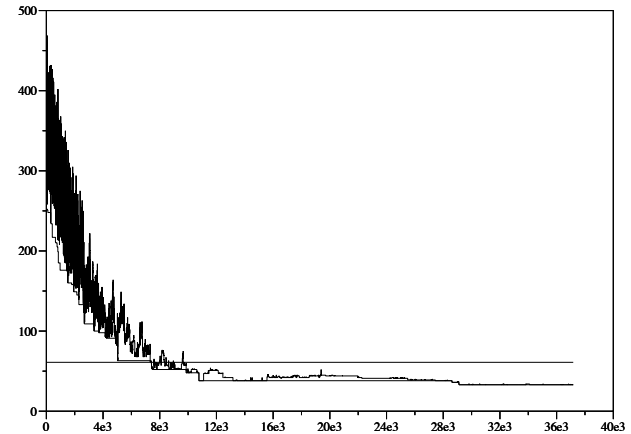
$\beta$	$\leq 5\%$	$]5\%, 6\%]$	$]6\%, 7\%]$
%	97.74	1.76	0.49

- Pour 20 des 23 instances avec  $\beta > 5\%$   
il a suffi de relancer l'algorithme 1.7  
fois, en moyenne.

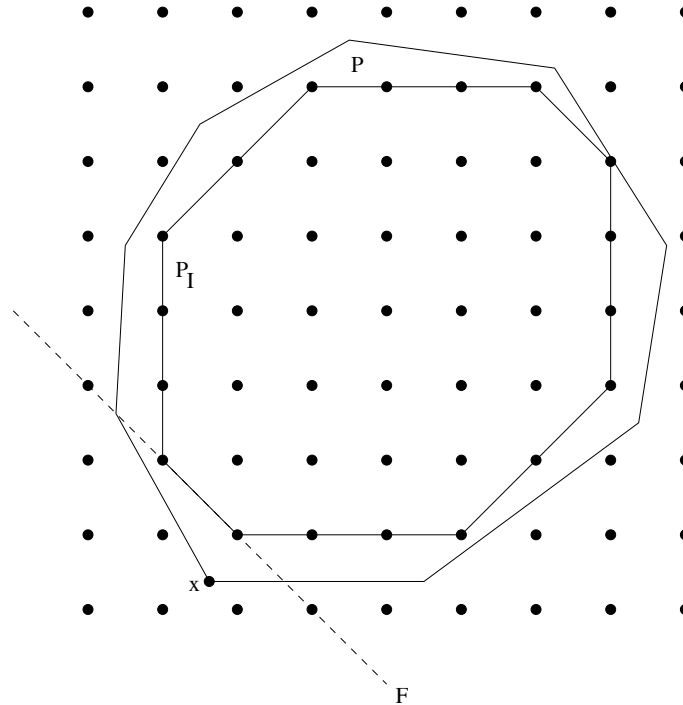
- Reste 3 instances ouvertes :

$ U $	$ M $	OPT	best $\beta$
13	36	?	5.65%
12	37	?	5.32%
10	24	?	5.90%

- D'où :  $0.9408 \leq \hat{\alpha} \leq 0.9676$ .



Convergence de l'algorithme.



$$\left\{ \begin{array}{l} \text{Minimiser } \sum_{m \in M} c_m \delta_{mm} \\ \text{s. l. c.} \\ \delta_{mm'} + \delta_{m'm} + \delta_{mm} + \delta_{m'm'} \geq 1, \forall \{m, m'\} \subseteq M \text{ (minicliques)} \\ \delta_{mm'} + \delta_{m'm} + \delta_{mm} \leq 1, m, m' \in M, m \neq m' \text{ (minimonocycles)} \\ \delta_{mm'} + \delta_{m'm''} - \delta_{mm''} \leq 1, m, m', m'' \in M, m \neq m' \neq m'' \neq m \text{ (transitivité)} \\ (1 - \delta_{mm})w_m \leq K_{t_m} + \sum_{m' \in S(t_m)} w_{m'}(\delta_{m'm'} + \delta_{m'm}) - \sum_{m' \in C(t_m) \setminus \{m\}} w_{m'} \delta_{m'm}, \forall m \in M \\ \delta_{mm'} \in \{0, 1\}, m, m' \in M \end{array} \right.$$

- Le polytope des sous-tournois sans circuit,  $P_{\text{PLO}}^n$  :
  - Pleine dimension.
  - Facettes :
    - $\delta_{ij} \geq 0$  ( $i \neq j$ ), contraintes de transitivité étendues.
    - $k$ -cliques (généralisent les minicliques),  $k$ -monocycles (généralisent les minimonocycles) et  $k$ - $l$ -bicyclettes.
    - Toute facette de  $P_{\text{PLO}}^n$  définit une facette de  $P_{\text{PLO}}^{n+1}$ .
- Le polytope des programmes de déplacement,  $P_{\text{PMP}}^M$  :
  - Pleine dimension (sous des conditions peu restrictives).
  - Facettes :
    - $\delta_{mm'} \geq 0$  ( $m \neq m'$ ), contraintes de transitivité étendues,  $k$ -cliques,  $k$ -monocycles et  $k$ - $l$ -bicyclettes (sous des conditions peu restrictives).
    - Contraintes de  $s$ - et de  $t$ -recouvrement.

- $k$ -cliques,  $k$ -monocycles,  $k$ - $l$ -bicyclettes :
  - Polynomiales en nombre (respectivement  $C_n^k$ ,  $nC_{n-1}^k$  et  $n^2C_{n-2}^kC_{n-k-2}^l$ ) pour  $k$  fixé,  $l$  fixé, donc polynomialement séparables.
  - Séparation  $NP$ -difficile au sens fort (par restriction au problème de la coupe maximum) dans le cas contraire.
- Contraintes de  $s$ - et de  $t$ -recouvrement :
  - Le problème de séparation se réduit à la résolution de  $2|M|$  sacs à dos et est donc pseudopolynomial.
  - Il en résulte que l'on peut venir à bout pseudopolynomialement d'une relaxation comprenant les  $O(|M|)^3$  contraintes de transitivité étendues, les  $k$ -cliques ( $k \leq k_0$ ,  $k_0$  fixé), les  $k$ -monocycles ( $k \leq k'_0$ ,  $k'_0$  fixé) et les contraintes de  $s$ - et de  $t$ -recouvrement, à l'aide de l'algorithme de l'ellipsoïde.

# Méthode de coupes et branchements [2] 23/29

- Obtention d’une « bonne » solution initiale à l’aide du recuit simulé précédemment présenté.
- Résolution de la relaxation comprenant :
  - Les  $|M|$  contraintes de capacité.
  - Les  $O(|M|^2)$  minicliques.
  - Les  $O(|M|^2)$  minimonocycles.
  - Les  $O(|M|^3)$  contraintes de transitivité étendues.
  - Les contraintes de  $s$ - et de  $t$ -recouvrement.
- Recherche arborescente.

$ U $	$ M $	$d(z_0)$	$d(\text{GLB})$	# nœuds
25	45	0.00%	0.33%	5
25	38	3.40%	0.00%	3
25	50	2.26%	0.24%	13
25	37	3.51%	0.00%	1
25	36	2.96%	0.00%	4
25	41	0.00%	0.00%	1
25	40	2.01%	0.22%	15
25	55	1.23%	0.00%	5
25	47	1.27%	0.00%	6
25	42	3.30%	0.00%	2
	43.1	1.99%	0.08%	

$ U $	$ M $	# inst.	# rés.	$\bar{d}(z_0)$	$\bar{d}(\text{GLB})$	$\bar{d}(z_f)$
15	23.2	10	10	1.92%	0.07%	0.00%
20	34.1	10	10	1.20%	0.33%	0.00%
25	43.1	10	10	1.99%	0.08%	0.00%
30	53.1	10	6	$\leq 2.01\%$	$\leq 1.05\%$	$\leq 0.74\%$
35	61.8	10	7	$\leq 2.38\%$	$\leq 0.98\%$	$\leq 0.86\%$
40	67.5	10	6	$\leq 2.23\%$	$\leq 0.48\%$	$\leq 0.43\%$
45	76.5	10	6	$\leq 2.50\%$	$\leq 1.15\%$	$\leq 1.02\%$
50	88.5	10	3	$\leq 2.69\%$	$\leq 1.80\%$	$\leq 1.72\%$
55	93.6	10	4	$\leq 3.22\%$	$\leq 1.34\%$	$\leq 1.27\%$
60	108.4	10	2	$\leq 2.80\%$	$\leq 2.52\%$	$\leq 2.52\%$
75	117.7	10	0	$\leq 3.69\%$	$\leq 3.69\%$	$\leq 3.69\%$
70	122.3	10	1	$\leq 3.32\%$	$\leq 3.02\%$	$\leq 3.00\%$
75	131.3	10	0	$\leq 4.08\%$	$\leq 4.08\%$	$\leq 4.08\%$
80	137.4	10	0	$\leq 3.30\%$	$\leq 3.30\%$	$\leq 3.30\%$



- Résultats théoriques :
  - Complexité et cas particuliers polynomiaux du problème de reconfiguration.
  - Introduction et étude de la notion d' $(\alpha, \beta)$ -acceptabilité.
  - Introduction et étude des polytopes  $P_{\text{PLO}}^n$  et  $P_{\text{PMP}}^M$ .
- Méthodes de résolution :
  - Une méthode arborescente relativement simple à implémenter pour la résolution exacte d'instances de taille moyenne.
  - Un algorithme de recuit simulé, viable dans un contexte temps réel embarqué, qui fournit *en pratique* des solutions « acceptables ».
  - Une méthode de coupes et branchements pour la résolution d'instances de taille conséquente.
- Ces méthodes permettent de résoudre toutes les instances réelles, de fait moins contraintes que nos instances de test.

- Répartition de charge sur le BSC et partitionnement [5].
- Configuration de cellules radio et couplage biparti [1].
- Répartition de cellules sur des liens MIC et *bin packing* [1].
- Partage équitable de ressources et théorie des systèmes électoraux [1].
- Consommation électrique d'une BTS et sac à dos max-min [1].
- Allocation dynamique de ressource de traitement d'appels et flots.
- Estimateur du maximum de vraisemblance des paramètres de modèles de croissance de fiabilité.
- Planification de reroutages dans les réseaux MPLS.
- Déconvolution/démodulation de *bursts* GSM et programmation quadratique.
- Algorithmes pour la résolution de Sudokus [10, 11].

- [1] R. SIRDEY, « Combinatorial optimization problems in wireless switch design », *JOR*, à paraître.
- [2] R. SIRDEY et H. KERIVIN, « A branch-and-cut algorithm for a resource-constrained scheduling problem », *RAIRO—Operations Research*, à paraître.
- [3] R. SIRDEY, J. CARLIER et D. NACE, « Approximate resolution of a resource-constrained scheduling problem », *Journal of Heuristics*, à paraître.
- [4] R. SIRDEY, J. CARLIER, H. KERIVIN et D. NACE, « On a resource-constrained scheduling problem with application to distributed systems reconfiguration », *European Journal of Operational Research*, à paraître.
- [5] R. SIRDEY, D. PLAINFOSSÉ et J.-P. GAUTHIER, « A practical approach to combinatorial optimization problems encountered in the design of a high availability distributed system », actes d'INOC p. 532-539, 2003.
- [6] R. SIRDEY et H. KERIVIN, « Polyhedral combinatorics of a resource-constrained scheduling problem part I—On the partial linear ordering polytope », *Mathematical Programming*, soumis.
- [7] H. KERIVIN et R. SIRDEY, « Polyhedral combinatorics of a resource-constrained scheduling problem part II—On the process move program polytope », *Mathematical Programming*, soumis.

- [8] R. SIRDEY, « Des solutions pour faire bonne figure », *La Recherche* n° 407, avril 2007.
- [9] R. SIRDEY, « Optimiser... En découpant des polyèdres », *L'Ouvert* n° 115, à paraître en 2007.
- [10] R. SIRDEY, « Sudokus et algorithmes de recuit », *Quadrature* n° 62, octobre-décembre 2006.
- [11] R. SIRDEY, « Sudokus et programmation linéaire », *Quadrature* n° 63, janvier-mars 2007.

**Merci !**