Le problème SAT à travers deux décennies d'investigations

Professeur Habiba DRIAS

MOTIVATION

- Les solveurs SAT ont connu une **évolution significative** ces dernières années
- L'utilisation de SAT connaît une croissance importante dans le **monde de l'industrie**
- Performances des stratégies de résolution
  - meilleure recherche basée sur le backtracking
  - meilleures structures de données
  - Nouvelles stratégies
  - Nouveaux paradigmes

Notation & Définitions

- Une instance SAT = Formule **CNF**
  - est une ***formule logique*** d'ordre 0 ou 1 souvent exprimée sous ***forme normale conjonctive ou CNF***
  - La conversion d'une formule logique quelconque en forme normale conjonctive est ***automatisable***
  - *(Davis M. & Putnam H., "A Computing Procedure for Quantification Theory", ACM 1960)*

PSAT: SAT Propositionnel

- **Littéral, clause, instance de SAT**
  - Si $V = \{x_1, x_2, \ldots, x_n\}$ est un ensemble de ***variables booléennes***
  - Un ***littéral*** est une variable booléenne simple ou complémentée
  - Une ***clause*** est une disjonction de littéraux
  - Une ***instance de SAT*** est une conjonction de clauses

Exemple d'une instance de SAT

- $-x_2$ x_4 x_5
- x_3 $-x_4$
- $-x_1$ x_3 $-x_5$

- donnée SAT constituée de 3 clauses formées à partir de 5 variables
- (00110) est un exemple d'instanciation (assignation, interprétation) des variables

- $(\neg x_2 \vee x_4 \vee x_5) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_3 \vee \neg x_5)$
  - ou bien
  - $(-x_2 + x_4 + x_5)\ (x_3 + -x_4)\ (-x_1 + x_3 + -x_5)$
- formule CNF logique correspondante

Problèmes associés à une donnée SAT

- Le problème de décision
  - • Donnée SAT
  - • Question : Existe-il une instanciation des variables qui satisfait toutes les clauses simultanément?

  - Réponse:
    - Oui si la donnée est satisfiable
    - Non si la donnée est contradictoire

Autres problèmes

- Le problème de recherche de solutions
  - Donnée SAT
  - Question : Trouver une ou plusieurs instanciations des variables qui satisfont toutes les clauses simultanément?
    - Réponse: √ une ou plusieurs interprétations des variables
- Le problème de dénombrement des solutions
  - Donnée SAT
  - Question : Déterminer le nombre des instanciations de variables qui satisfont toutes les clauses simultanément?
    - Réponse: √ Le nombre de solutions

Le problème de la satisfiabilité maximale ou MAX-SAT

- Quand la donnée est contradictoire, une question qu'on pourrait se poser est de connaître le nombre maximum de clauses qui pourraient être satisfaites simultanément. Ce problème est appelé MAX-SAT:
  - Le problème de décision:
    - Instance SAT, un entier positif k
    - Question : Existe-il une instanciation des variables qui satisfait p clauses simultanément où p>=k?
  - Le problème d'optimisation:
    - Instance SAT
    - Question : Trouver une instanciation des variables qui satisfait le maximum de clauses simultanément?

La satisfiabilité maximale pondérée ou MAX-W-SAT

- Une donnée SAT ***pondérée***:

- √ est une instance dans laquelle ***un poids est associé à chaque clause***

- √ Question: Trouver une assignation des variables qui maximise la ***somme des poids des clauses*** qui sont satisfaites en même temps

Le problème SAT incrémental

- √ Une donnée SAT ***S*** satisfiable
- √ une clause ***C***
- √ Question: La donnée ***{S,C}*** est-elle satisfiable?

Applications

- Exemples de domaines d'applications
  - Bases de données
  - VLSI
  - Recherche Opérationnelle
  - Intelligence Artificielle
  - **Tout domaine qui utilise la logique**

En bases de données

- Le problème SAT
  - Détection des incohérences dans les bases de données
- Le problème MAX-SAT
  - Elimination des incohérences dans les bases de données
- Le problème SAT incrémental
  - Détection des incohérences dans les bases de données déductives

En Intelligence Artificielle

- Démonstration de théorèmes
  - Solveur SAT
    - *Cook S.A., "The Complexity of Theorem proving procedures", ACM symposium on Theory of Computing, 1971*
- Résolution de problèmes
  - Solveur SAT
- Raisonnement automatique
  - Solveur SAT

- Les solveurs SAT sont utilisés quasiment dans toutes les disciplines de l'I.A.

Les Solveurs SAT

- Les plus efficaces reposent sur les méthodes suivantes:
  - Recherche "retour arrière" ou Backtracking
  - propagation de contraintes booléennes
  - stratégies de recherche
    - backtracking aléatoire
    - méta-heuristiques
  - structures de données efficaces
    - queues, watched literal, …
- Font l'objet de **compétitions internationales**
- **Exemples: BerkMin, GRASP, SATO ...**

SAT et la complexité de calcul

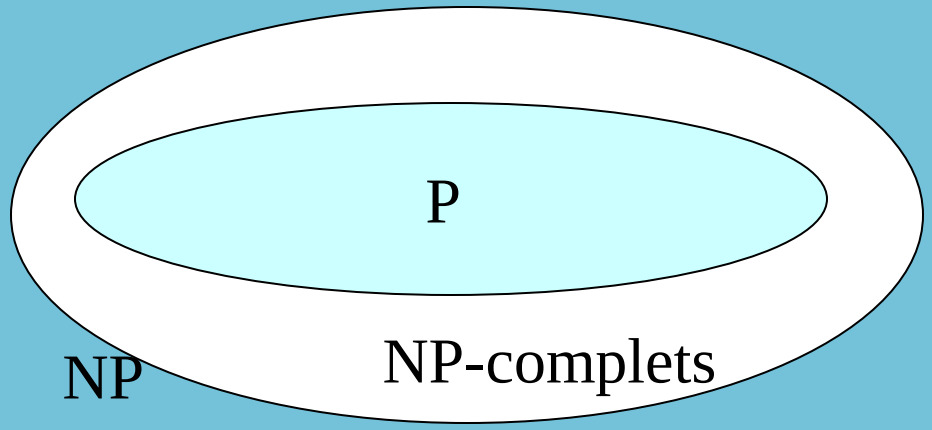
- La Théorie de la NP-complétude a démarré avec l'énoncé du premier théorème sur SAT
  - **Théorème: Le problème de satisfiabilité est NP-complet**
    - *Cook S.A., "The Complexity of Theorem proving procedures", ACM symposium on Theory of Computing, 1971*
    - *Garey M. & Johnson D.S., **Computers and Intractability** : a Guide to the theory of NP-completeness, Freeman 1979*
- La démonstration est basée sur l'interprétation des caractéristiques des problèmes NP à l'aide de la logique

SAT: Ancêtre des problèmes NP-complets

Les classes de problèmes NP et P

- La classe **NP** regroupe les problèmes résolus par des algorithmes non déterministes
  - Un algorithme non déterministe a l'ossature suivante:
    - générer une instanciation quelconque
    - vérifier que l'instanciation est solution ou pas
- La classe **P** englobe les problèmes résolus par des algorithmes déterministes polynomiaux

NP = P ?



- La zone NP-P est-elle vide?
  Pas de réponse jusqu'à aujourd'hui
- Investigations intensives pour explorer la zone NP-P

Un problème est-il polynomial ou NP-complet?

- Un algorithme est acceptable que si sa complexité est non exponentielle
- Démarche
  - Ecrire un algorithme pour le problème
  - Calculer sa complexité (du pire cas)
  - Si (non exponentiel) alors le problème $\in$ P
  - Sinon ?
- ?
  - Chercher un autre algorithme qui soit polynomial
  - sinon penser à démontrer qu'il est NP-complet

Définition formelle d'un problème NP-complet

- Un problème Π est NP-complet si
  - 1 $\Pi \in$NP
  - 2 il existe une transformation polynomiale entre un autre problème NP-complet et Π

Autres classes de complexité

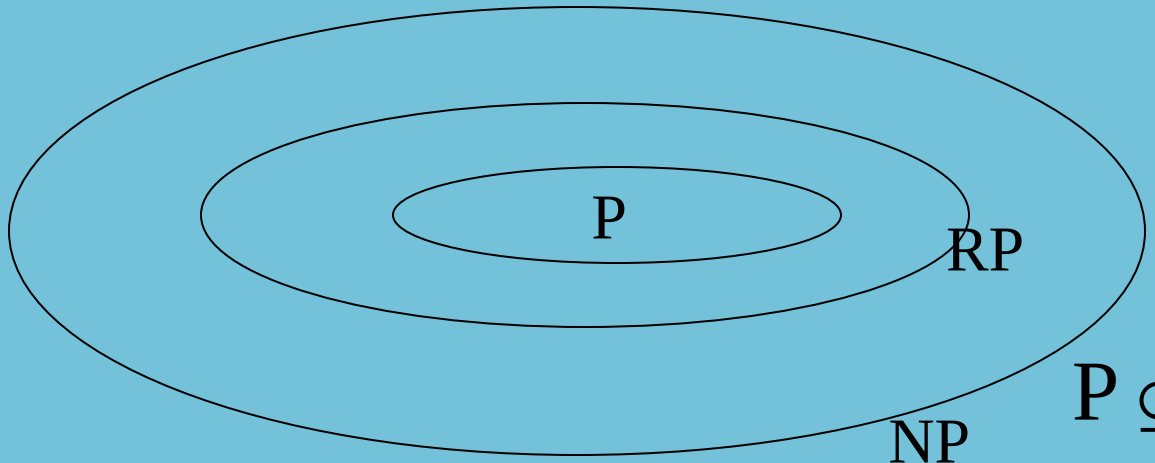
- pour mieux explorer la zone NP-P, Cook a préconisé l'utilisation
  - des probabilités (en particulier)
  - et du parallélisme
- Dans une conférence présentée à l'occasion de l'obtention de son award en 1983

Naissance de la classe RP

- La classe **RP** englobe les problèmes qui possèdent un algorithme probabiliste pour leur résolution

- Un **algorithme probabiliste** est un algorithme qui
  - contient une instruction de génération de données aléatoires
  - et produit
    - une réponse correcte
    - ou incorrecte mais avec une probabilité d'erreur très faible ($< \varepsilon$)

RP = P?



$P \subseteq RP \subseteq NP$

Pas de réponse à la question jusqu'à aujourd'hui

Au delà des problèmes NP-complets

- Les problèmes NP-difficiles
  - problèmes complexes de recherche et d'optimisation
- Les problèmes #-complets
  - problèmes complexes de dénombrement
- Les problèmes APX et PTAS
  - les problèmes d'approximation [Maffioli 92]

Variantes de SAT

- Comme SAT est NP-complet, pour contourner la difficulté, les investigations se sont tournées vers **certaines variantes** du problème

- Une variante de SAT = SAT + **restriction**

Exemples de variantes de SAT

- r-SAT = donnée SAT où toutes les clauses sont de longueur égale à r (≤ r dans certaines littératures)
- r,s-SAT = donnée r-SAT où chaque variable apparaît au plus s fois
- Horn-SAT : donnée SAT avec des clauses de Horn
  - Dans une clause de Horn, une seule variable est complémentée

Quelques Résultats sur SAT

- Sont polynomiales les variantes suivantes:
  - 2-SAT
  - Horn-SAT
    - la version de Prolog II est basée sur les clauses de Horn

Quelques Résultats sur SAT

- Sont NP-complets les variantes suivantes:
  - r-SAT lorsque $r \geq 3$
  - MAX-SAT
  - MAX-2-SAT
  - MAX-Horn-SAT
  - MAX-W-SAT

Problèmes d'optimisation

- Sont NP-difficiles les problèmes d'optimisation suivants:
  - MAX-SAT
  - MAX-2-SAT
  - MAX-Horn-SAT
  - MAX-W-SAT

La procédure DPP

- ***Davis M. & Putnam H., "A Computing procedure for Quantification Theory", ACM 1960***
- Une procédure pendant longtemps étudiée
- C'est un algorithme exact avec:
  - Un comportement exponentiel
  - Une complexité moyenne polynomiale sur une grande plage de données
    - **[Goldberg A. 1979], [Franco J. 1986],**
    - **[Purdom P. & Brown C.A. 1987]**
    - **et d'autres ...**

Autres Algorithmes

- [Iwama K. 1989]
  - complexité moyenne sur une plus grande plage de données
- [Dubois 0. 1991]
  - Nombre moyen de solutions
- [Drias H. 1992, 1997, 2001]
  - Lois de distribution de probabilités du nombre de solutions
    - *Journal of Information Science and Engineering, Vol 8 (1992)*
  - Généralisation de la formule du nombre moyen de solutions
    - *Computers and Artificial Intelligence (1997)*
  - complexité moyenne sur une plus grande plage de données
    - *Journal of Combinatorial Mathematics and Combinatorial Computing (2001)*

Les Solveurs SAT et la complexité de calcul

- SAT est appliqué dans le secteur industriel
  - Automatisation des composants électronique
  - vérification formelle des systèmes matériels et logiciels
  - et d'autres ...
- Des solveurs SAT, on attend la résolution d'instances de problèmes:
  - qui dépassent des **centaines de milliers, voire des millions de variables**
  - avec des **dizaines de millions de clauses**
  - Ils doivent être capables de démontrer la non satisfiabilité, la complétude étant le but principal

SAT et l'Intelligence Artificielle

- Démonstration de théorèmes
  - Un théorème peut être converti en une formule logique
  - Un solveur SAT est équivalent à un démonstrateur de théorèmes
- Résolution de problèmes
  - Utilisation des Solveurs SAT

La résolution de problèmes

- Elle est concernée par les problèmes complexes comme les problèmes NP-complets
- Les problèmes NP-complets sont de mieux en mieux résolus grâce:
  - à l'existence d'une pléthore de méta-heuristiques
  - aux performances des nouvelles machines
- Les solveurs SAT sont basés le plus souvent sur des heuristiques

Les Heuristiques les plus connues pour SAT et MAX-SAT

- John1 (Johnson 1974)
- John2 //
- G-SAT (Selman et al 1993)
- GRASP (Resende et all 1997)

Quelques méta-heuristiques existantes

- Ascent Descent Methods (Hansen and Jaumard 1990)
  - Simulated Annealing
  - Steepest Ascent Mildest Descent
- Genetic algorithm (Frank 1994 …)
- Reactive Tabu Search (Battiti and Protasi 1997)
- Tabu Search (Mazure et al 1997)
- Guided local search (Mills et all 1999)
- Scatter Search (Drias et all 2001)
  - *'Scatter search with random walk strategy for solving hard MAX-W-SAT problems'**, in proc of IEA-AIE'2001, LNAI 2070**, (2001)*
  - *'Randomness in Heuristics : An Experimental Investigation for the Maximum Satisfiability Problem'**, JCMCC** (2001)*

An overview of Scatter Search

- Scatter Search is a recent evolutionary approach (Glover et all 2000)
- It is a population-based meta-heuristic
- It is used for the first time to solve SAT and MAX-W-SAT
- Main Specificities:
  - Scatter search does not resort to randomization
  - The population is kept scattered all along the process
  - Combination of solutions is based on an integer programming formulation
  - More than two solutions can be combined at a time

Scatter Search framework

- Generate an initial population P
- **while not** Stop-Condition **do**
  - Initialize the reference set with the solutions selected to be combined
  - Generate new solutions by applying the combination process
  - Improve new solutions quality
  - Insert new solutions in population with respect to quality and dispersion criteria
- **end while**

Modeling SAT and MAX-W-SAT by Scatter Search

- Solution representation
  - a solution is a string of bits representing Boolean variables values
- Initial population
  - a diversification generator is used
    - for each h <= n-1 and solution (x_1, …,x_n)
    - $x'_1 = 1 - x_1$
    - $x'_{1+h*k} = 1 - x_{1+h*k}$
  - a generated solution is improved by means of a heuristic

- Diversification generator
  - distance(x,y) is computed to measure the solutions diversity
  - It is designed to be the sum of the absolute values of the differences between the corresponding bits
- Reference set
  - is created with the best solutions of the initial population in terms of :
    - quality
    - diversity

- Combination method
  - the value to attribute to a variable is the value of the variable of the solution that maximizes the weights sum of the satisfied clauses
  - example:
    - x_1 $-x_3$ x_4 3
    - x_3 $-x_4$ $-x_5$ 4
    - $-x_2$ x_3 $-x_5$ 2

- C((01001), (01101), (01010)) = 01100.

Scatter Search with Random walk strategy

- The random walk strategy consists in introducing a noise in the search in order to deviate it from its usual rules
- with a probability smaller than p execute the Scatter search
- with a probability greater than p invert a bit chosen at random

Procedure RWSS-SAT;

- **begin**
- S = seed; { generated randomly }
- **For** (Iter = 1 To maxIter) **do**
- **begin**
- RN = random(0..1);
- **If** RN < p
  - **then** SS-SAT(S)
  - **Else** invert a bit of S that increases the number of satisfied clauses;
- S = x*; { x* being the new solution }
- **End;**
- **End;**

Experimentations

- We have implemented in C on a personal Pentium computer
  - SS-SAT
  - RWSS-SAT
- Types of instances
  - Johnson Benchmark: real-life problems
  - 800 to 1000 clauses
  - http://www.research.att.com/~mgcr/data/index.html

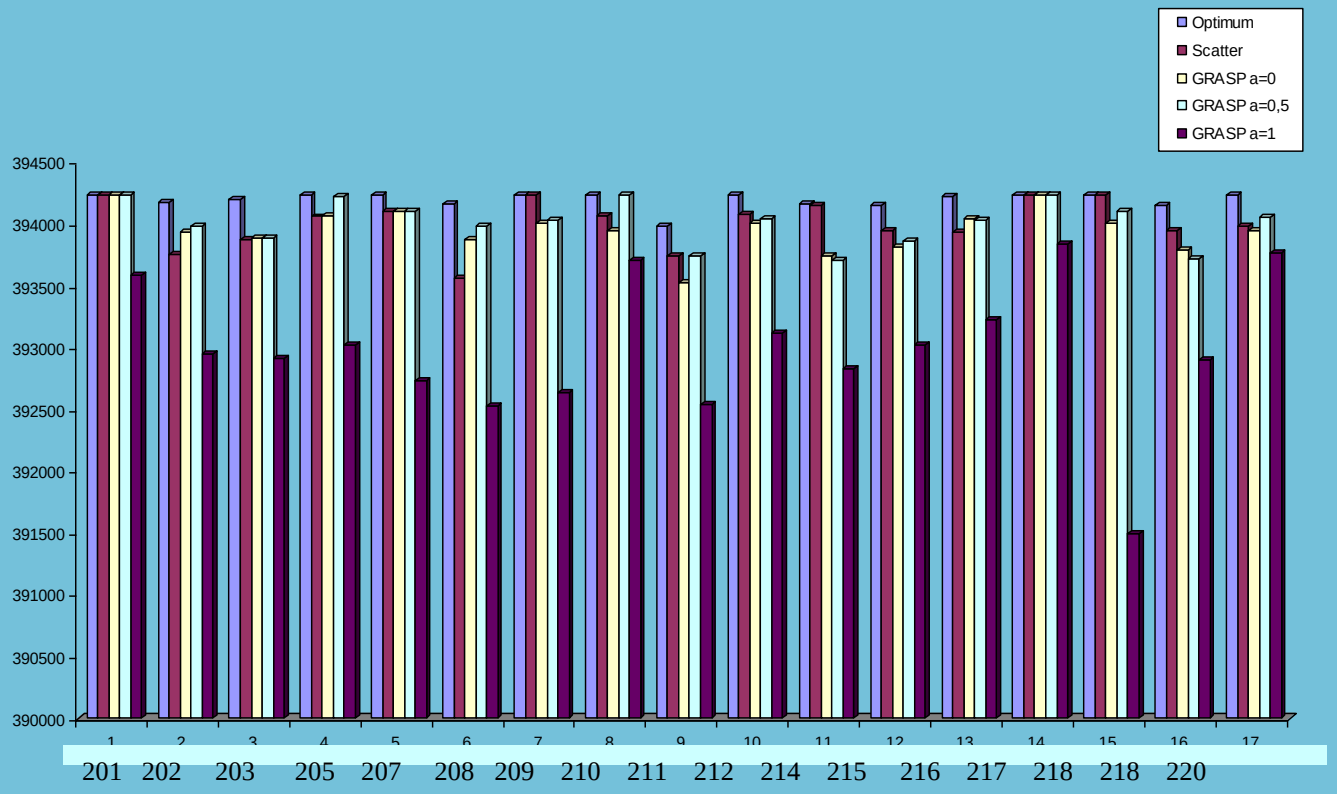
Numerical results

- Parameters setting
  - for SS-SAT
    - P-size = 150
    - b1 = 5
    - b2 = 5
    - max-iter = 5
  - for RWSS-SAT
    - P-size = 150
    - b1 = 5
    - b2 = 5
    - max-iter = 30
    - p = 0.5

Experimental results comparing SS-SAT and sequential GRASP

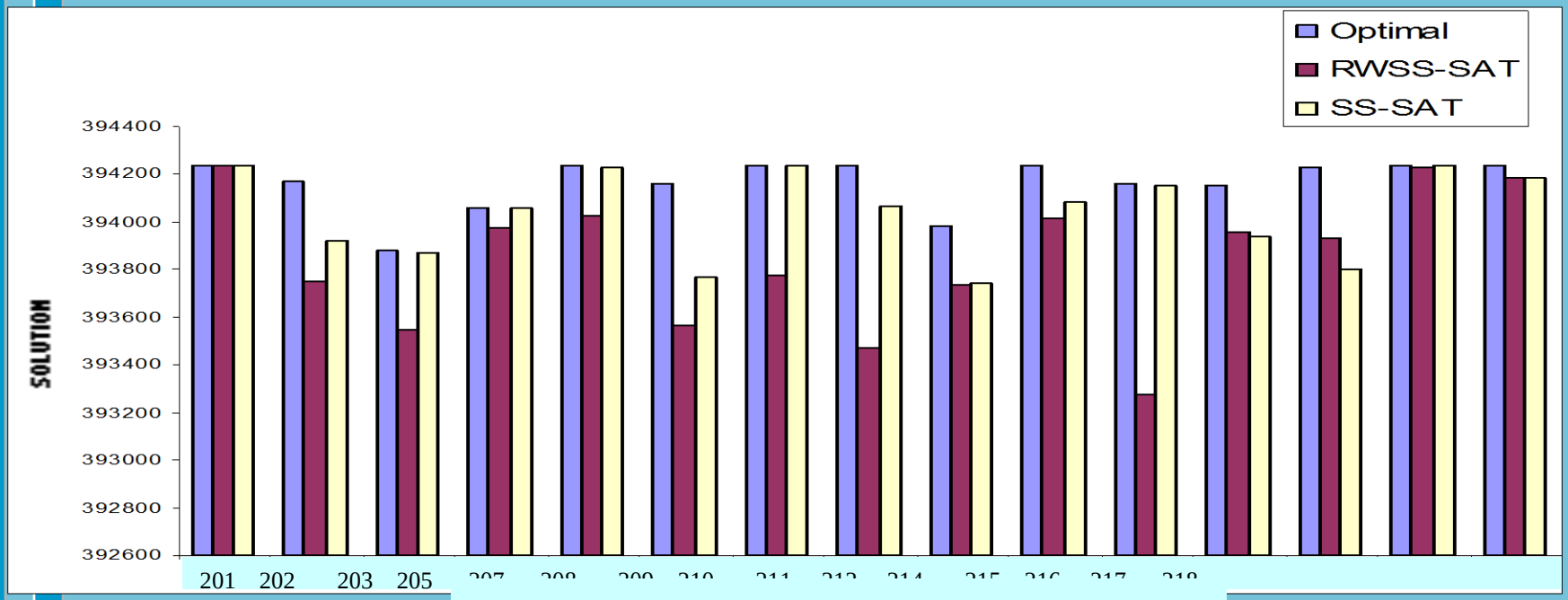
instance	Optimal solution	GRASP		SS-SAT	
		solution	time	solution	time
Jnh01	420925	420632	1525.2	420892	203.99
Jnh10	420840	420463	964.4	420479	901.47
Jnh11	420753	420398	1330.1	420141	439.82
Jhn12	420925	420518	564.1	420701	250.65
Jnh13	420816	420592	567.0	420716	930.12
Jhn14	420824	420491	982.6	420491	298.22
Jhn15	420719	420429	899.2	420632	624.63
Jhn16	420919	420809	18.7	420889	716.51
Jhn17	420925	420712	1062.8	420794	401.95
Jhn18	420795	420289	560.5	420404	129.64
Jhn19	420759	420307	287.3	420330	403.10
Jnh201	394238	394154	1099.6	394238	161.72
Jnh202	394170	393680	261.2	393924	269.99
Jnh203	393881	393446	1352.3	393875	294.10
Jnh205	394063	393890	1163.5	394060	203.37
Jnh207	394238	394030	41.8	394228	313.58
Jnh208	394159	393859	890.1	393771	137.19
Jnh209	394238	393959	1740.0	394238	463.31
Jnh210	394238	393950	219.3	394067	394.46
Jhn211	393979	393529	281.8	393742	408.04
Jhn212	394238	394011	171.5	394082	758.00
Jhn214	394163	393737	1272.2	394152	1159.15
Jhn215	394150	393818	350.8	393942	202.51
Jhn216	394226	394042	1215.7	393806	181.43
Jhn217	394238	394232	438.8	394238	799.40
Jhn218	394238	394009	825.2	394189	166.47
Jhn219	394156	393792	1308.3	393800	530.35
Jhn220	394238	393951	1055.1	393985	597.70
Jhn301	444854	444612	347.6	444842	1267.63
Jhn302	444459	443906	46.6	443895	698.77
Jhn303	444503	444063	1046.7	444223	437.91
Jhn304	444533	444310	142.5	444533	1175.98
Jhn305	444112	444112	1465.8	443594	463.33
Jhn306	444838	444603	1003.0	444515	604.53
Jhn307	444314	443836	972.3	443662	288.14
Jhn308	444724	444215	607.9	444250	768.57
Jhn309	444578	444273	564.6	444483	662.58

Experimental results comparing SS-SAT and sequential GRASP



Johnson Benchmarks

Experimental results comparing SS-SAT and RWSS-SAT



Johnson Benchmarks

Comments

- SS-SAT outperforms the **sequential version of GRASP**, which is yet a very efficient heuristic for MAX-SAT
- The running time for **SS-SAT** is more interesting than that for **GRASP**
- When the random noise strategy is introduced in SS-SAT no increase in performance is observed
- The running time for **SS-SAT** is even better than for **RWSS-SAT**

Ouvrages sur SAT

- Que ce soit en I.A. ou en Complexité de calcul, des chapitres d'ouvrages récents sont consacrés au problème SAT
- Exemples:
  - Battiti R.
    - *Handbook on SAT, 1997*
  - Pardalos P.M. & Resende M.G.
    - *Handbook of Applied Optimization, oxford University Press, 2002*
    - *Handbook of Massive Data Set, Kluwer Academic, 2002*
  - Nilson J.
    - *Artificial Intelligence, Kluwer Academic, 2001*
  - Barthélémy J.P.
    - *Complexité algorithmique, Masson ,1992*

Une conférence rien que pour SAT

- Pour la première fois
  - SAT'2002
    - organisé par Pr John FRANCO
    - à l'université de Cincinatti
- Http:// gauss.ececs.uc.edu/Conferences/SAT2002/
- Et depuis la conférence se tient tous les ans