# Scaling and Probabilistic Smoothing: Dynamic Local Search for Unweighted MAX-SAT

Dave Tompkins & Holger Hoos
University of British Columbia

AI 2003. Halifax, NS
June 13th

# Satisfaction

- Where do we go for Dinner?
- Jacques wants Poutine (P v ¬J)
- Bill wants Steak (S v ¬B)
- Dave wants Lobster or Steak or Poutine or Thai (L v S v P v T v ¬D)
- Restaurant X has Lobster & Steak: (L v ¬X) (S v ¬X)
- (P v ¬J) (S v ¬B) (L v S v P v T v ¬D) (L v ¬X) (S v ¬X)

# SAT & MAX-SAT

- $\underbrace{(L \vee S \vee P \vee \overset{\text{literal}}{T} \vee \neg D)}_{\text{clause}}$
- Objective: Find an assignment (L=1,S=0, etc..) that SATisfies:
  - ALL clauses (SAT) or
  - as many clauses as possible (MAX-SAT)

# Stochastic Local Search for SAT

- variable assignment → clauses are **satisfied** or **unsatisfied**

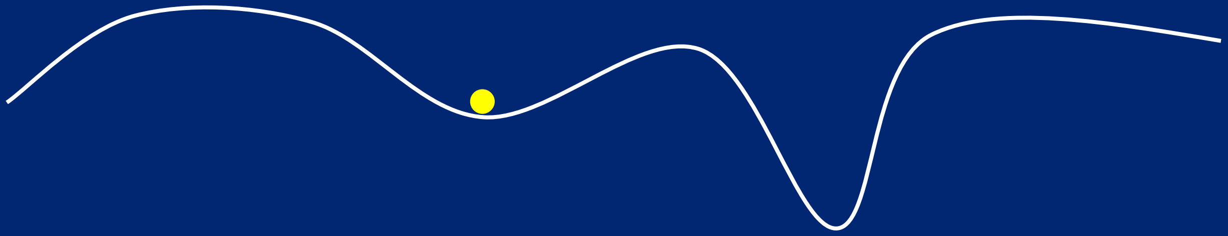
- [101001011] → **(1) (1) (1) (0) (1) (0) (0) (0)**

- Choose a variable to flip

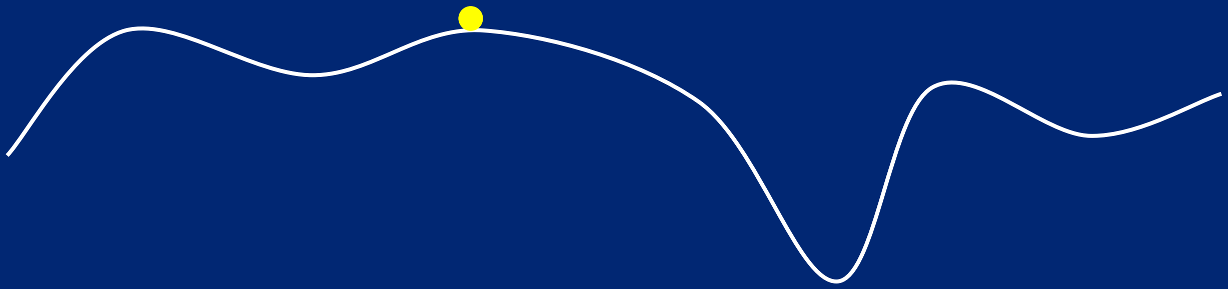
- [1010**1**1011] → **(1) (0) (1) (1) (1) (1) (0) (0)**
- [10100101**0**] → **(1) (0) (1) (1) (0) (1) (1) (1)**
- [1**1**1001011] → **(1) (0) (0) (1) (1) (1) (1) (1)**

# Dynamic Local Search (DLS)

- Minimisation problem: Search Space



- Dynamic Local Search: "warp" the space



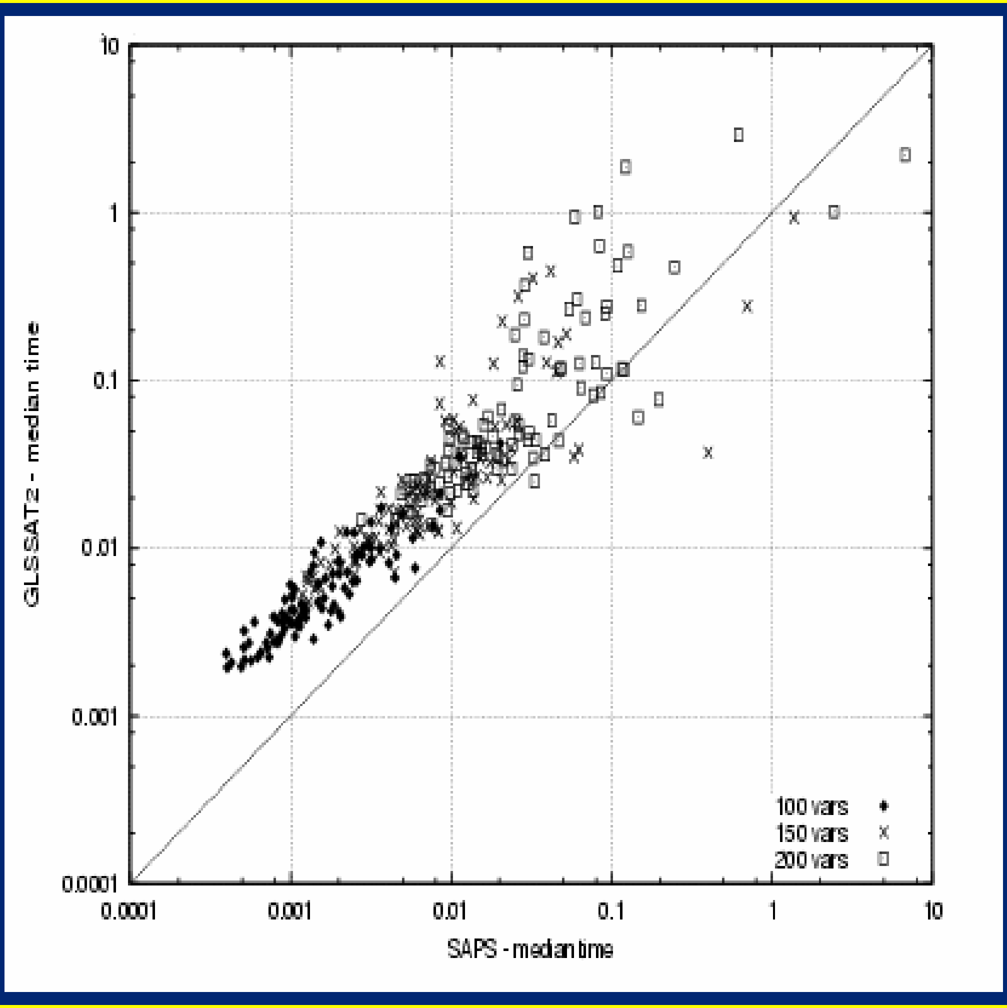
# Scaling & Probabilistic Smoothing

- Assign all clauses a clause penalty: clp (start = 1)
- When a Local Minimum is encountered:
  - Scaling Step:
    All unsatisfied clauses are scaled by $\alpha$
    $\mathrm{clp}_i \leftarrow \alpha \cdot \mathrm{clp}_i$
  - Smoothing Step:
    All clauses are smoothed ($\rho$) towards the mean
    $\mathrm{clp}_i \leftarrow \mathrm{clp}_i \cdot \rho + \mathrm{clp}_{avg} \cdot (1- \rho)$
- Smoothing Step performed with probability $P_{smooth}$

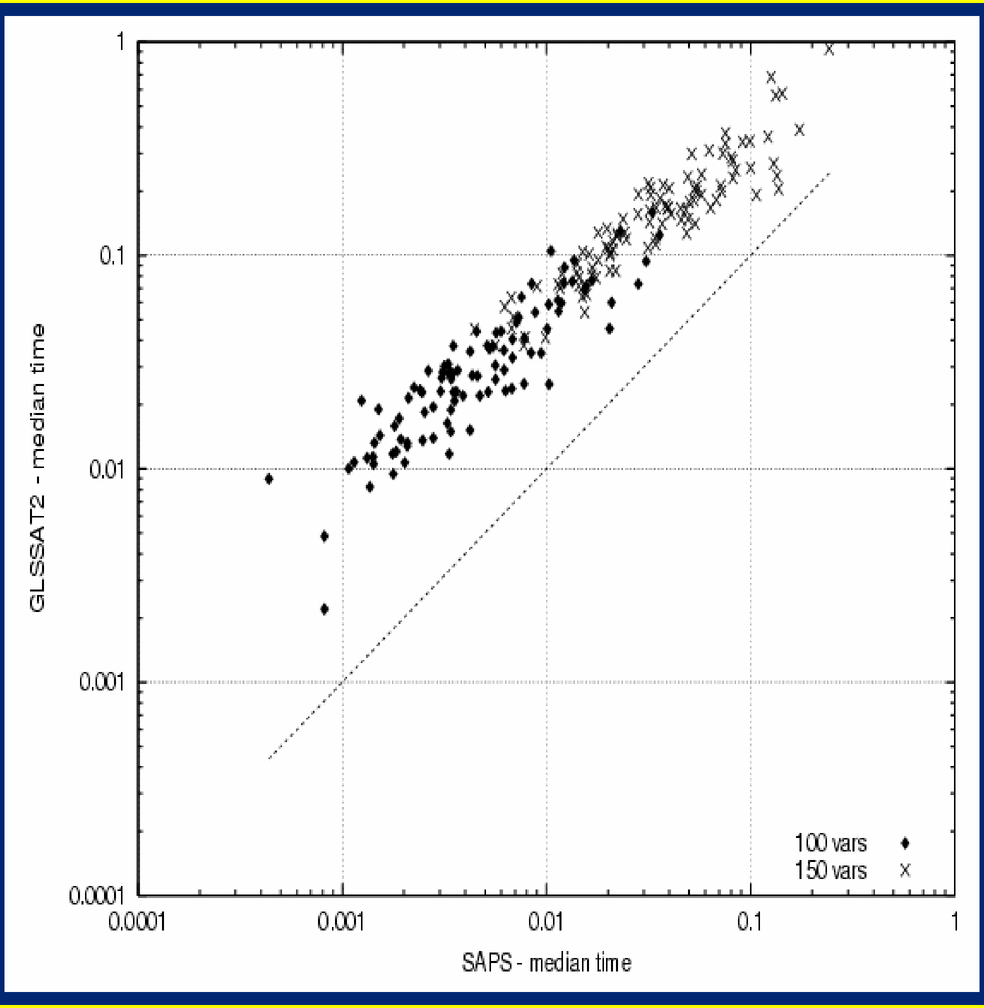
# SAPS For MAX-SAT

- SAPS: Scaling ($\alpha$) and smoothing ($\rho$)
- SAPS amongst State-of-the-Art for SLS SAT
- Objective: Apply SPAS to unweighted MAX-SAT
- Weighted MAX-SAT: Each clause has a weight $w_i$
- Compare against another DLS algorithm: GLS2 (Mills & Tsang): current state-of-the-art MAX-SAT

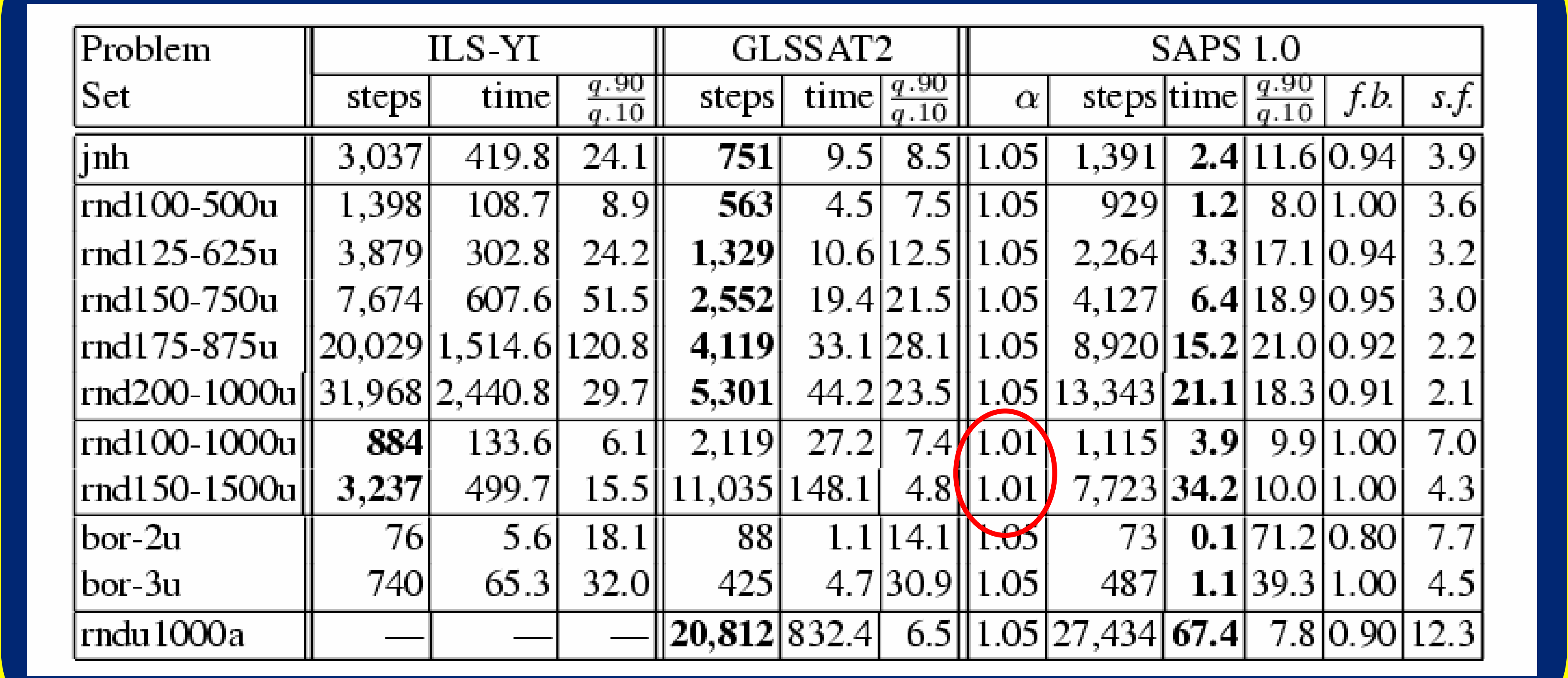
# Slightly over-constrained



# Heavily over-constrained



# Results on Unweighted MAX-SAT



| Problem Set | ILS-YI | | | GLSSAT2 | | | SAPS 1.0 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | steps | time | $\frac{q_{.90}}{q_{.10}}$ | steps | time | $\frac{q_{.90}}{q_{.10}}$ | $\alpha$ | steps | time | $\frac{q_{.90}}{q_{.10}}$ | *f.b.* | *s.f.* |
| jnh | 3,037 | 419.8 | 24.1 | **751** | 9.5 | 8.5 | 1.05 | 1,391 | **2.4** | 11.6 | 0.94 | 3.9 |
| rnd100-500u | 1,398 | 108.7 | 8.9 | **563** | 4.5 | 7.5 | 1.05 | 929 | **1.2** | 8.0 | 1.00 | 3.6 |
| rnd125-625u | 3,879 | 302.8 | 24.2 | **1,329** | 10.6 | 12.5 | 1.05 | 2,264 | **3.3** | 17.1 | 0.94 | 3.2 |
| rnd150-750u | 7,674 | 607.6 | 51.5 | **2,552** | 19.4 | 21.5 | 1.05 | 4,127 | **6.4** | 18.9 | 0.95 | 3.0 |
| rnd175-875u | 20,029 | 1,514.6 | 120.8 | **4,119** | 33.1 | 28.1 | 1.05 | 8,920 | **15.2** | 21.0 | 0.92 | 2.2 |
| rnd200-1000u | 31,968 | 2,440.8 | 29.7 | **5,301** | 44.2 | 23.5 | 1.05 | 13,343 | **21.1** | 18.3 | 0.91 | 2.1 |
| rnd100-1000u | **884** | 133.6 | 6.1 | 2,119 | 27.2 | 7.4 | 1.01 | 1,115 | **3.9** | 9.9 | 1.00 | 7.0 |
| rnd150-1500u | **3,237** | 499.7 | 15.5 | 11,035 | 148.1 | 4.8 | 1.01 | 7,723 | **34.2** | 10.0 | 1.00 | 4.3 |
| bor-2u | 76 | 5.6 | 18.1 | 88 | 1.1 | 14.1 | 1.05 | 73 | **0.1** | 71.2 | 0.80 | 7.7 |
| bor-3u | 740 | 65.3 | 32.0 | 425 | 4.7 | 30.9 | 1.05 | 487 | **1.1** | 39.3 | 1.00 | 4.5 |
| rndu1000a | — | — | — | **20,812** | 832.4 | 6.5 | 1.05 | 27,434 | **67.4** | 7.8 | 0.90 | 12.3 |

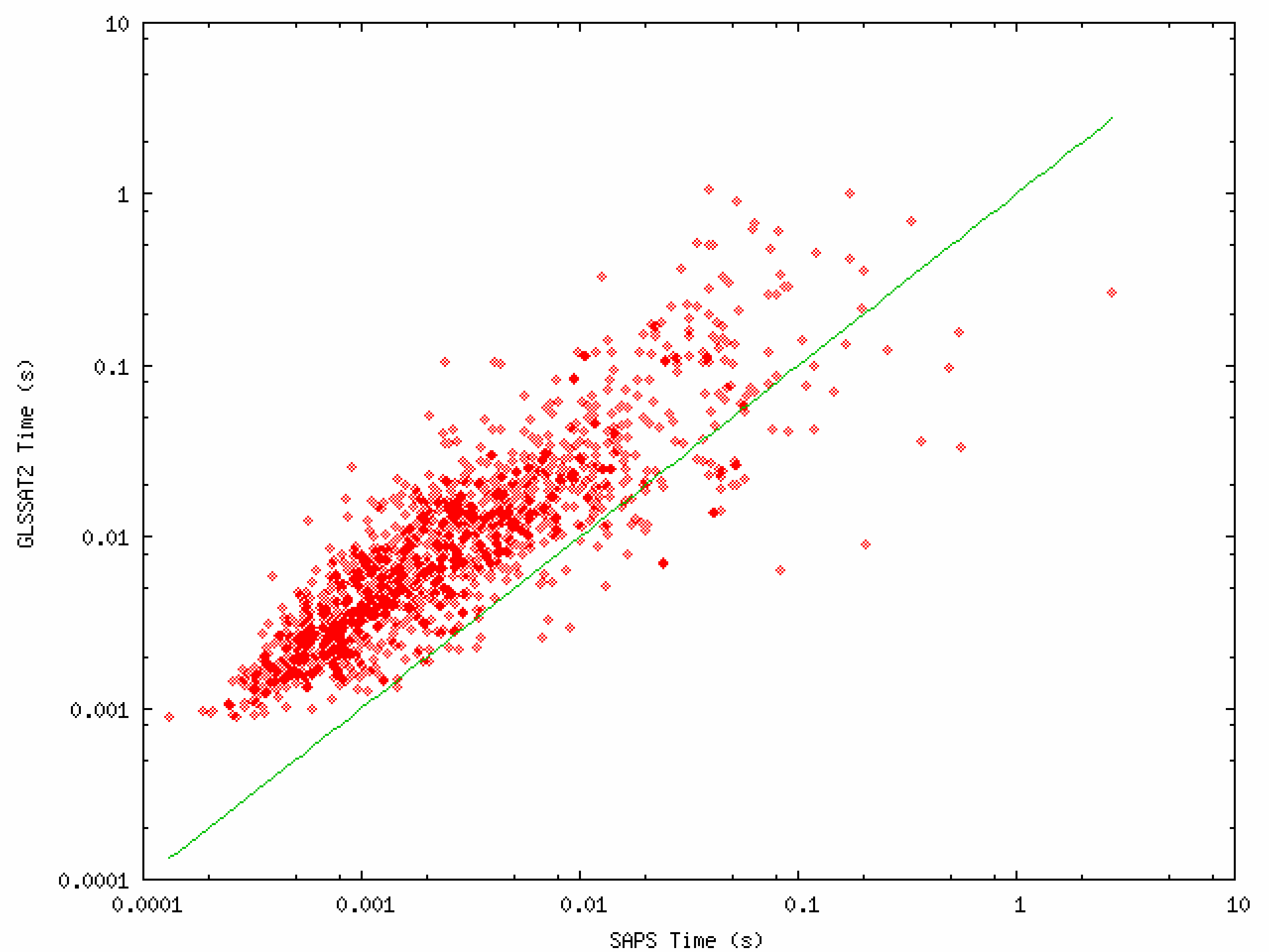
# SAPS for Unweighted MAX-SAT

- Typical optimal $\alpha_{\text{MAX-SAT}}$ is much smaller (1.05) than typical optimised $\alpha_{\text{SAT}}$ (1.3)
- Key difference between SAT & MAX-SAT
  - Unsatisfied clauses in the solution!
  - Global minimum can “move”
- Harder instances require an even smaller value of $\alpha$ (1.01)
- Possible correlation to search space characteristics?

# Moving to Weighted MAX-SAT

- We have two distinct clause "weights"
  - – MAX-SAT weights
  - – SAPS Clause Penalties
- We need some mechanism to combine them
  - – $w_i + clp_i$ (or something like $w_i + k \cdot (clp_i / clp_{max})$
  - – $w_i * clp_i$
  - – GLS approach: $util(i) = w_i / (1 + clp_i)$

# Weighted MAX-SAT Results



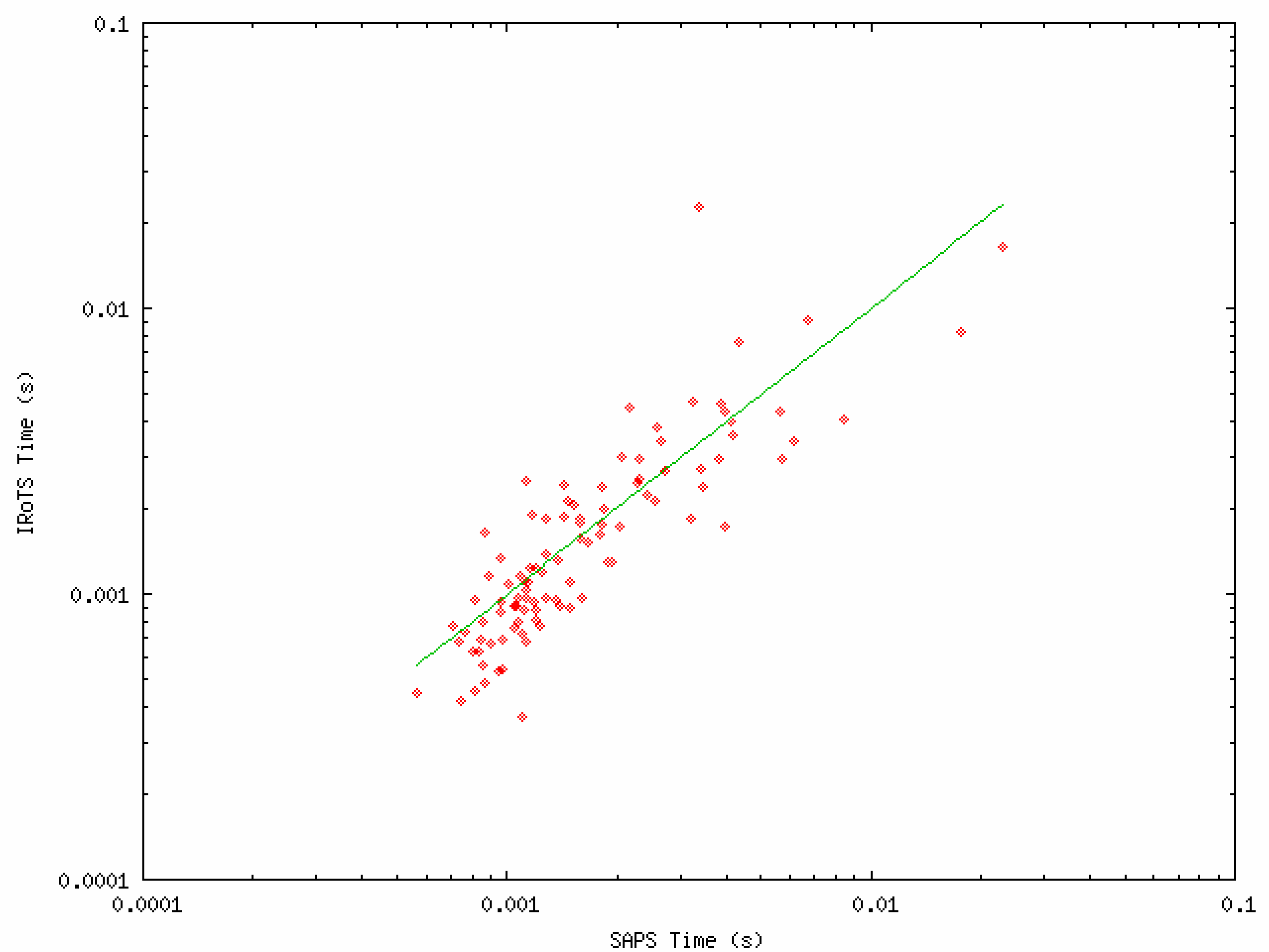
# Conclusions

- SAPS is effective on unweighted MAX-SAT
- Only small changes in default parameters were necessary
- Great potential for weighted MAX-SAT

# Future Work - Hypotheses

- For MAX-SAT, a scaling α-reactive scheme would be more effective than a smoothing ρ-adaptive scheme
- A more clever approach to combining MAX-SAT weights and SAPS clause penalties may exist

# IRots vs. SAPS


IRoTS Time (s)
SAPS Time (s)