

# Cooperation control in Parallel SAT Solving

N. Lazaar, Y. Hamadi, S. Jabbour, M. Sebag



# Parallel SAT Solving

## Decentralized resolution

- ▶ Each core: conflict-directed clause learning
- ▶ Cooperation: each core sends the learned clauses to other cores
- ▶ Why? additional clauses help pruning the search space.

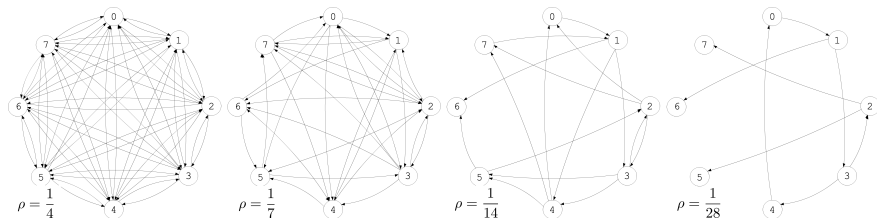
## Previous work

Hamadi et al. 09

- ▶ Controlling the length of the shared clauses (TCP/IP congestion avoidance)

# Limitations

Does not scale up when the number of cores increases.



## Position of the problem

- ▶ Dynamically control the topology of the network
- ▶ This paper: density  $\rho$  is fixed

# BESS

## Bandit Ensemble for parallel SAT Solving

### Core tasks

- ▶ Design the reward
- ▶ Adjust the decision schedule wrt internal SAT schedule

### BESS structure

- ▶ Each receiver core
- ▶ selects  $n$  emitter cores  $n = 1/2 \# \text{ cores}$

# Designing the reward of an emitter core

**Reward(emitter):** sum of reward(shared clauses)

## I. Global clause rewards

- ▶ Size-based: clause of length  $s$  removes  $2^{N-s}$  instances

$$r(c) = -\log 1 - 2^{-s}$$

- ▶ Literal-block distance  
each literal (decision level) produces unit propagations  
LBD: difference between highest and lowest decision levels in the clause literals
- ▶ Mixtures of the above

FAIL

# Designing the reward of an emitter core, 2

## II. Receiver-specific clause rewards

- ▶ Literals  $\ell$  are associated their activity  $a(\ell)$   
# (their assignment  $\rightarrow$  failure)

$$r(c) = \frac{1}{c} \sum_{\ell \in c} \text{sigmoid} \left( \frac{a(\ell)}{a_{max}} \right)$$

# BESS Algorithm

In each core, independently

- ▶ Maintain a reward threshold
- ▶ Update the reward of alive emitters relaxation
- ▶  $\Pr(\text{removing emitter}) = \Pr(\text{emitter reward} < \text{threshold})$
- ▶ Turns (oldest) sleeping emitters into alive ones to achieve  $n$  alive emitters at all time.

# Experimental setting

## Platforms

- \* 8-core Intel Xeon, 16 GB RAM, 2.33GHz
- \* 32-core AMD Opteron Proc. 6136, 64GB RAM, 2.4GHz.

## SAT instances

SAT-Challenge 2012.

588 SAT+UNSAT instances.

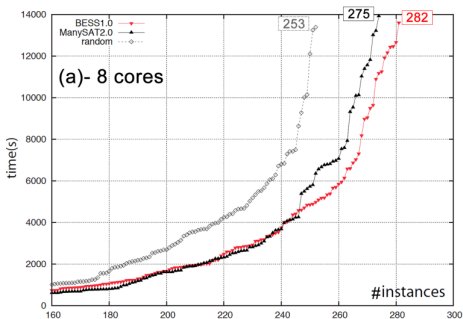
## Parameters

- \* CPU time limit = 30mn CPU per core
- \* Shared clause limit size: 8
- \* Alive emitters:  $1/2$  nb of cores.

## Baseline

Random selection of alive emitters in each time step.

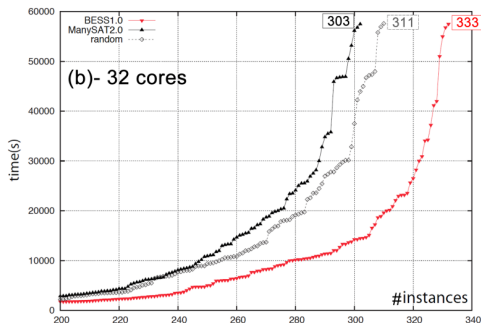
# Results on 8 cores



## Comments

- ▶ Bess slightly improves on ManySAT 2.0 for difficult problems

# Results on 32 cores



## Comments

- ▶ Random improves on ManySAT 2.0 (confirms scalability issue)
- ▶ Bess improves on Random and ManySAT 2.0 solves the first 300 pbs in 20,000 s. versus 50,000 s.

# Perspectives

1. Adjust the *number* of emitters for each core
2. Adjust the clause length limit
3. Share information among cores to speed-up cooperation, enforce diversification.