# The Fractal Dimension of SAT Formulas

C. Ansótegui[1] M.L. Bonet[2] **J. Giráldez-Cru**[3] J. Levy[3]

[1]DIEI - Univ. de Lleida

[2]LSI - UPC

[3]**IIIA-CSIC**

IJCAR 2014
July 19, 2014

## Known Facts from SAT Community

- **Random** and **industrial** formulas: **distinct nature**.
  - SAT competitions: different tracks.
- SAT solvers **specialize**.
- Many **very large industrial instances solved efficiently** by modern SAT solvers (**CDCL**).
  - Good performance: ability to exploit some **hidden structure**.

# SAT Instances

- **Random k-CNF**:
  - Its definition is clear.
  - Generate $k$-CNF of $n$ vars and $m$ clauses:

```
for i in 1..m
   Select randomly k literals among n
     with random polarity
```

  - Theoretical point of view.

- **Industrial CNF**:
  - Problems encodings from real-world applications.
  - No precise definition: crypto, bmc, scheduling, planning, ...
  - Heterogeneity.

# SAT Instances

- **Random k-CNF**:
  - Its definition is clear.
  - Generate $k$-CNF of $n$ vars and $m$ clauses:

    ```
    for i in 1..m
       Select randomly k literals among n
         with random polarity
    ```
  - Theoretical point of view.

- **Industrial CNF**:
  - Problems encodings from real-world applications.
  - No precise definition: crypto, bmc, scheduling, planning, ...
  - Heterogeneity.

# [Some] Open Questions in SAT

- **Open Question #1**: What is exactly the **structure** of industrial formulas?
- **Open Question #2**: How SAT solvers (can) **exploit** this structure?

## Complex Networks

- The **classical Erdös-Rényi model**:
  - Generate a graph of $n$ nodes and $m$ edges:

    ```
    for i in 1..m
       Select randomly 2 nodes among n
    ```
  - These networks cannot be used for representing many real-world networks.

- **Real-world networks**:
  - **Features**: Clustering coefficient, Modularity, ...
  - **Models**: Small-world, Scale-free, ...
  - Methods of **generation**: Preferential attachment, ...

# Complex Networks

- The **classical Erdös-Rényi model**:
  - Generate a graph of $n$ nodes and $m$ edges:
    ```
    for i in 1..m
       Select randomly 2 nodes among n
    ```
  - These networks cannot be used for representing many real-world networks.

- **Real-world networks**:
  - **Features**: Clustering coefficient, Modularity, ...
  - **Models**: Small-world, Scale-free, ...
  - Methods of **generation**: Preferential attachment, ...

# Complex Networks vs SAT

- **Erdös-Rényi graphs**:

```
for i in 1..m
  Select randomly 2 nodes among n
```

- **Random k-CNF**:

```
for i in 1..m
  Select randomly k literals among n
   with random polarity
```

- **Real-world networks**: features, models, methods of generation.
- **Industrial CNF**: ?

# Complex Networks vs SAT

- **Erdös-Rényi graphs**:
  ```
  for i in 1..m
    Select randomly 2 nodes among n
  ```
- **Random k-CNF**:
  ```
  for i in 1..m
    Select randomly k literals among n
     with random polarity
  ```
- **Real-world networks**: features, models, methods of generation.
- **Industrial CNF**: **?**

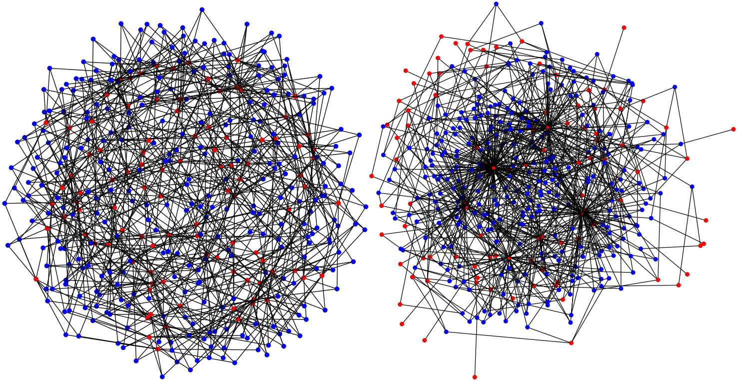
# [Some] Open Questions in SAT

- **Open Question #1**: What is exactly the **structure** of industrial formulas?
- **Open Question #2**: How SAT solvers (can) **exploit** this structure?
- Many works in terms of complex networks trying to answer these questions.

# Previous Work (I)

- **Open Question #1**: What is exactly the **structure** of industrial formulas?

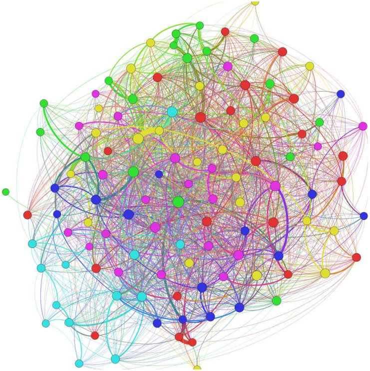
  Scale-free Structure [Ansótegui, Bonet, Levy. CP2009]

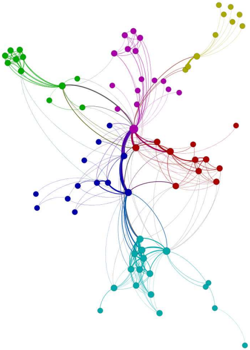


# Previous Work (II)

- **Open Question #1**: What is exactly the **structure** of industrial formulas?

Community Structure [Ansótegui, Giráldez-Cru, Levy. SAT2012]





# Previous Work (III)

- **Open Question #1**: What is exactly the **structure** of industrial formulas?
- **Open Question #2**: How SAT solvers (can) **exploit** this structure?
  - Centrality & Branching vars [Katsirelos, Simon. CP2012]
  - Parallel SAT Solving [Sonobe, Kondoh, Inaba. SAT2014]
  - LBD & Runtime Prediction [Newsham, Ganesh, Fischmesiter, Audemard, Simon. SAT2014] *Best Paper Award*
  - ...

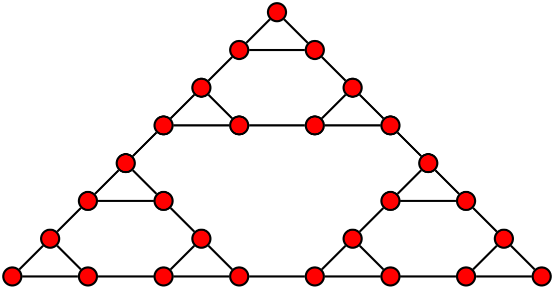
# Motivations

- **Analysis** of the structure of industrial SAT instances.
- **Generators** of more realistic industrial-like SAT formulas.
- (Possible) **improvements** in SAT solving techniques.

# Outline

# Intuition

A graph has **fractal dimension** (it is self-similar) if it keeps the same shape after *rescaling*.



# Intuition

A graph has **fractal dimension** (it is self-similar) if it keeps the same shape after *rescaling*.

0.5,0.151.5,1.152.5,0.152.5,2.153.5,3.154.5,2.154.5,0.155.5,1.156.5,

# Intuition

A graph has **fractal dimension** (it is self-similar) if it keeps the same shape after *rescaling*.

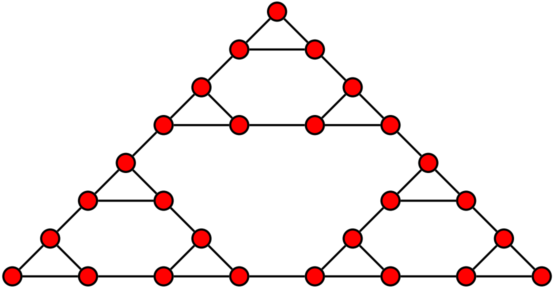
1.5,0.553.5,2.555.5,0.550.5,0.151.5,1.152.5,0.152.5,2.153.5,3.154.5,

# Computing the Fractal Dimension (I)

- **[Def.]** A **circle** of radius $r$ and center $c$ is a subset of nodes of the graph such that the distance between any of them and the node $c$ is strictly smaller than $r$.

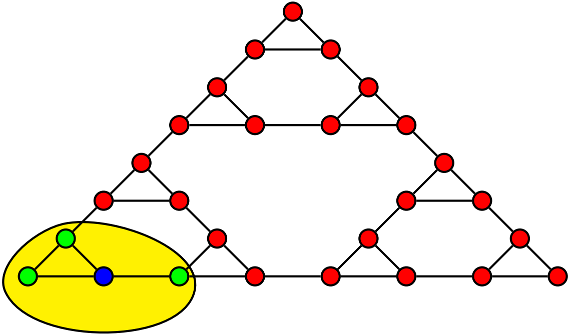
- **[Def.]** Let $\mathbf{N(r)}$ be the minimum number of circles of radius $r$ required to cover a graph.
  - $N(1) = n$
  - $N(d^{max} + 1) = 1$

# Computing the Fractal Dimension (II)



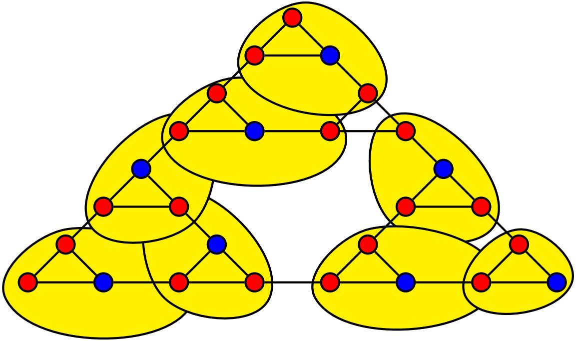
| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (II)



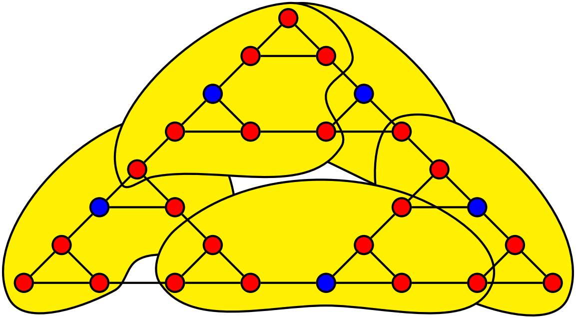
| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (II)



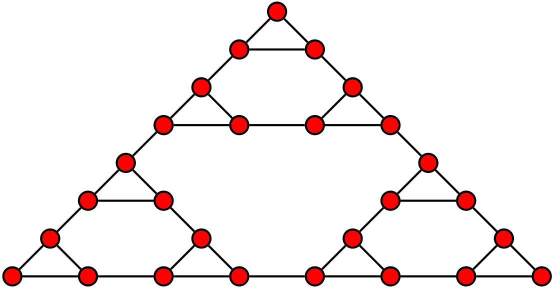
| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | 8 | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (II)



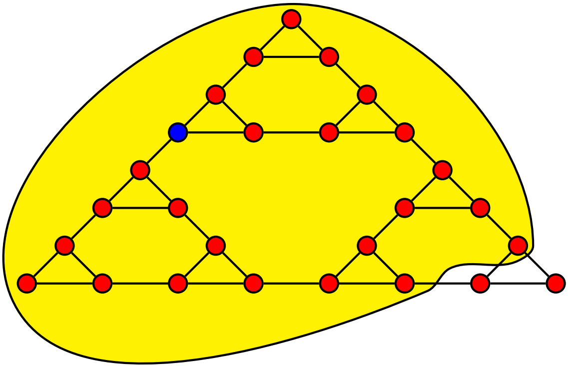
| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | 8 | |
| 3 | 5 | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (II)



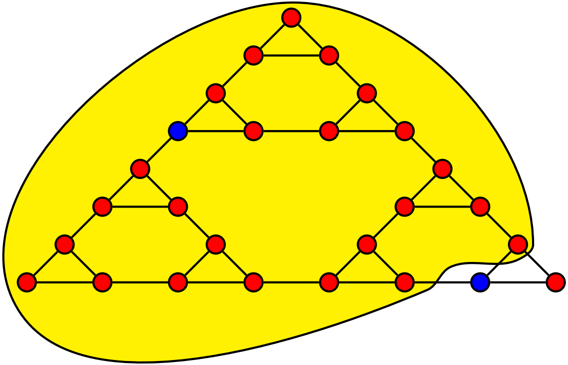
| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | 8 | |
| 3 | 5 | |
| 4 | 3 | |
| 5 | 2 | |
| 6 | 2 | |
| 7 | | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (II)



| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | 8 | |
| 3 | 5 | |
| 4 | 3 | |
| 5 | 2 | |
| 6 | 2 | |
| 7 | | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (II)



| $r$ | $N(r)$ | |
|---|---|---|
| 1 | 27 | #nodes |
| 2 | 8 | |
| 3 | 5 | |
| 4 | 3 | |
| 5 | 2 | |
| 6 | 2 | |
| 7 | 2 | |
| 8 | 1 | $d^{max} = 7$ |

# Computing the Fractal Dimension (III)

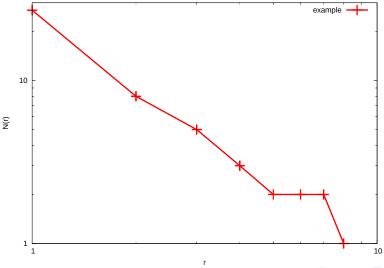
- **[Def.]** (*Hausdorff*) A graph has the **self-similarity** property if the function $N(r)$ decreases polynomially.
- I.e. $N(r) \sim r^{-d}$, for some value $d$.
- In the case, we call $d$ the **dimension** of the graph.

| $r$ | $N(r)$ |
|---|---|
| 1 | 27 |
| 2 | 8 |
| 3 | 5 |
| 4 | 3 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 1 |

# Computing the Fractal Dimension (III)

- **[Def.]** (*Hausdorff*) A graph has the **self-similarity** property if the function $N(r)$ decreases polynomially.
- I.e. $N(r) \sim r^{-d}$, for some value $d$.
- In the case, we call $d$ the **dimension** of the graph.

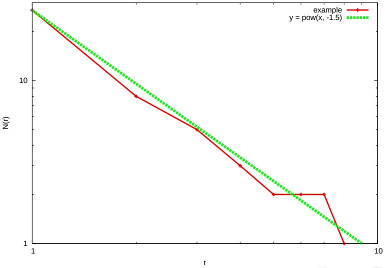
| $r$ | $N(r)$ |
|---|---|
| 1 | 27 |
| 2 | 8 |
| 3 | 5 |
| 4 | 3 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 1 |



# Computing the Fractal Dimension (III)

- **[Def.]** (*Hausdorff*) A graph has the **self-similarity** property if the function $N(r)$ decreases polynomially.
- I.e. $N(r) \sim r^{-d}$, for some value $d$.
- In the case, we call $d$ the **dimension** of the graph.

| $r$ | $N(r)$ |
|---|---|
| 1 | 27 |
| 2 | 8 |
| 3 | 5 |
| 4 | 3 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 1 |



# Computing the Fractal Dimension (IV)

- **[Lemma]** Computing the function $N(r)$ is **NP-hard**.
  - Reducing *GraphCOL* to $N(2)$.
- **Burning** algorithms:
  - More efficient algorithms (greedy).
  - Approximate upper bounds of $N(r)$.
  - Select the circle that covers (burns) the maximal number of uncovered (unburned) nodes.
  - Further approximations needed to make the algorithms of practical use in very large graphs.
- The **Burning by Node Degree (BND)** algorithm.

# Computing the Fractal Dimension (IV)

- **[Lemma]** Computing the function $N(r)$ is **NP-hard**.
  - Reducing *GraphCOL* to $N(2)$.
- **Burning** algorithms:
  - More efficient algorithms (greedy).
  - Approximate upper bounds of $N(r)$.
  - Select the circle that covers (burns) the maximal number of uncovered (unburned) nodes.
  - Further approximations needed to make the algorithms of practical use in very large graphs.
- The **Burning by Node Degree (BND)** algorithm.

# Computing the Fractal Dimension (IV)

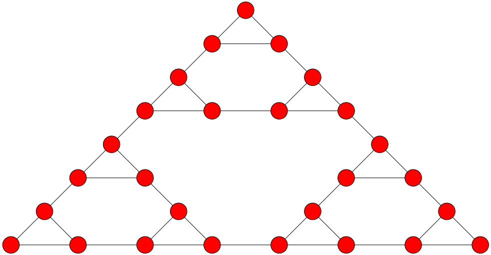
- **[Lemma]** Computing the function $N(r)$ is **NP-hard**.
  - Reducing *GraphCOL* to $N(2)$.
- **Burning** algorithms:
  - More efficient algorithms (greedy).
  - Approximate upper bounds of $N(r)$.
  - Select the circle that covers (burns) the maximal number of uncovered (unburned) nodes.
  - Further approximations needed to make the algorithms of practical use in very large graphs.
- The **Burning by Node Degree (BND)** algorithm.

# The Burning by Node Degree (BND) Algorithm

**Algorithm 1** Burning by Node Degree (BND)

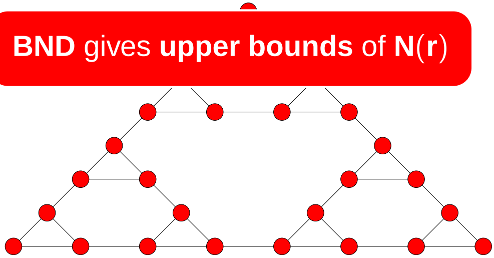
```
Input: Graph G = (V, E)
Output: vector[int] N
N[1] := |V|
int i := 2
while N[i − 1] > connectedComponents(G) do
    vector[bool] burned(|V|)
    N[i] := 0
    burned :={false, ..., false}
    while existsUnburnedNode(burned) do
        c := highestDegreeUnburnedNode(G, burned)
        S := circle(c, i)
        for all x ∈ S do
            burned[x] :=true
        end for
    end while
    i := i + 1
end while
```

# Example



| $r$ | $N^{real}(r)$ | $N^{BND}(r)$ |
|---|---|---|
| 1 | 27 | 27 |
| 2 | 8 | 9 |
| 3 | 5 | 6 |
| 4 | 3 | 3 |
| 5 | 2 | 2 |
| 6 | 2 | 2 |
| 7 | 2 | 2 |
| 8 | 1 | 1 |

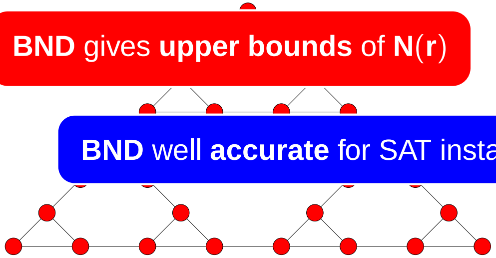
# Example



**BND** gives **upper bounds** of **N**(**r**)

| $r$ | $N^{real}(r)$ | $N^{BND}(r)$ |
|---|---|---|
| 1 | 27 | 27 |
| 2 | 8 | 9 |
| 3 | 5 | 6 |
| 4 | 3 | 3 |
| 5 | 2 | 2 |
| 6 | 2 | 2 |
| 7 | 2 | 2 |
| 8 | 1 | 1 |

# Example



**BND** gives **upper bounds** of **N(r)**

**BND** well **accurate** for SAT instances

| $r$ | $N^{real}(r)$ | $N^{BND}(r)$ |
|---|---|---|
| 1 | 27 | 27 |
| 2 | 8 | 9 |
| 3 | 5 | 6 |
| 4 | 3 | 3 |
| 5 | 2 | 2 |
| 6 | 2 | 2 |
| 7 | 2 | 2 |
| 8 | 1 | 1 |

# Fractal Dimension vs Diameter

- Determines the maximal radius $r^{max}$.
- Related to the diameter: $r^{max} \leq d^{max} \leq 2r^{max}$
- **Diameter**:
  - **Dependent** on the size of the graph.
  - Quite **expensive** to compute in practice.
- **The fractal dimension**:
  - **Independent on the size.** Families with similar $N(r)$ function shape.
  - It can be **computed more efficiently** than the diameter.

# Fractal Dimension vs Diameter

- Determines the maximal radius $r^{max}$.
- Related to the diameter: $r^{max} \leq d^{max} \leq 2r^{max}$
- **Diameter**:
  - **Dependent** on the size of the graph.
  - Quite **expensive** to compute in practice.
- **The fractal dimension**:
  - **Independent on the size**. Families with similar $N(r)$ function shape.
  - It can be **computed more efficiently** than the diameter.

# Fractal Dimension vs Diameter

- Determines the maximal radius $r^{max}$.
- Related to the diameter: $r^{max} \leq d^{max} \leq 2r^{max}$
- **Diameter**:
  - **Dependent** on the size of the graph.
  - Quite **expensive** to compute in practice.
- **The fractal dimension**:
  - **Independent on the size**. Families with similar $N(r)$ function shape.
  - It can be **computed more efficiently** than the diameter.

We propose the use of the **Fractal Dimension**

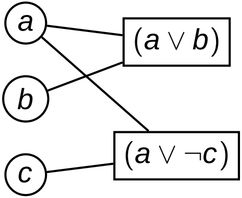
# Outline

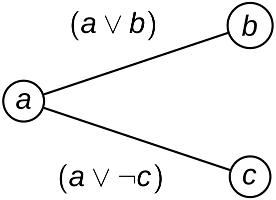
# SAT Formulas as Graphs

$$\sigma = (a \vee b) \wedge (a \vee \neg c)$$

**Clause-Variable Incidence Graph (CVIG)**

**Variable Incidence Graph (VIG)**



# The Relation between VIG and CVIG

| VIG | CVIG |
|---|---|
| $N(r)$ | $N^b(r)$ |

**[Lemma]**

$$\text{If } N(r) \sim r^{-\mathbf{d}} \Longrightarrow N^b(r) \sim r^{-\mathbf{d}}$$
$$\text{If } N(r) \sim e^{-\beta r} \Longrightarrow N^b(r) \sim e^{-\frac{\beta}{2} r}$$

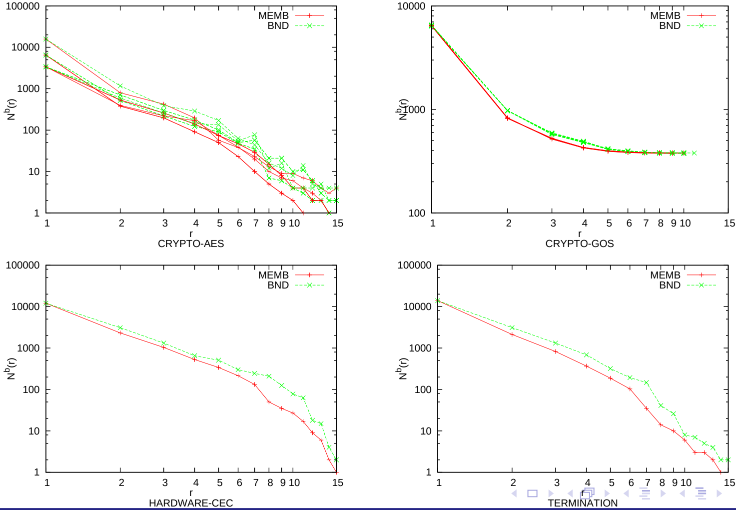
# The Accuracy of the BND Algorithm (I)

| | BND | MEMB[1] |
|---|---|---|
| #solved | 300 | 17 |
| av. of runtime | 0.11sec | 10min 7.2sec |
| $N^b(r)$ | Very similar values | |

Set: 300 industrial instances of the SAT Competition 2013

[1] [Song et al. Journal of Statical Mechanics (2007)]

# The Accuracy of the BND Algorithm (II)



# Known Results for Random 2CNF Formulas

- Phase transition point at $m/n = 1$.

- VIG's of random 2CNF formulas = Erdös-Rényi graphs.
- Percolation threshold at $m/n = 0.5$.
  - In this point, self-similar ($d = 2$).
  - Above this point $N(r)$ decays exponentially.

- To the best of our knowledge, there is **no known result for random 3CNF instances**.

# Known Results for Random 2CNF Formulas

- Phase transition point at $m/n = 1$.

- VIG's of random 2CNF formulas $=$ Erdös-Rényi graphs.
- Percolation threshold at $m/n = 0.5$.
  - In this point, self-similar ($d = 2$).
  - Above this point $N(r)$ decays exponentially.

- To the best of our knowledge, there is **no known result for random 3CNF instances**.

# Known Results for Random 2CNF Formulas

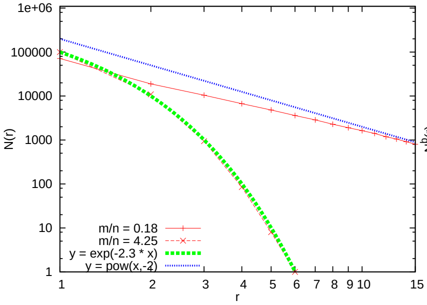
- Phase transition point at $m/n = 1$.

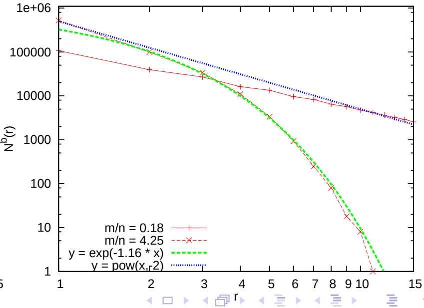
- VIG's of random 2CNF formulas $=$ Erdös-Rényi graphs.
- Percolation threshold at $m/n = 0.5$.
  - In this point, self-similar ($d = 2$).
  - Above this point $N(r)$ decays exponentially.

- To the best of our knowledge, there is **no known result for random 3CNF instances**.

# Random 3CNF Formulas

- Experimentally, $N(r)$ (and $N^b(r)$) only depends on the clause/variable ratio $m/n$ (and not on the number of variables $n$).
- Phase transition point ($m/n \approx 4.25$):
  - $N(r) \sim e^{-2.3r}$ and $N^b(r) \sim e^{-1.16r}$
  - Hence, the decay of CVIG is just half of the decay of VIG (as expected)
- (Experimentally) Percolation threshold at $m/n \approx 0.17$, $d = 2$



# Industrial SAT Formulas (I)

- Analysis of the SAT Competition 2013 (300 instances).
- Most industrial SAT instances are **self-similar**: $2 \leq d \leq 4$.
- Most families have homogeneous behaviors
- The size of the formulas does not affect the value of the dimension of the family (same slope for all instances).

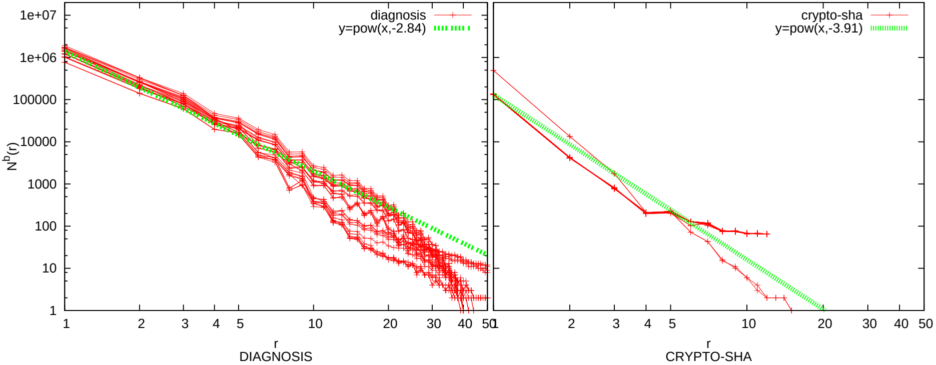
# Industrial SAT Formulas (I)

- Analysis of the SAT Competition 2013 (300 instances).

- Most industrial SAT instances are **self-similar**: $2 \leq d \leq 4$.

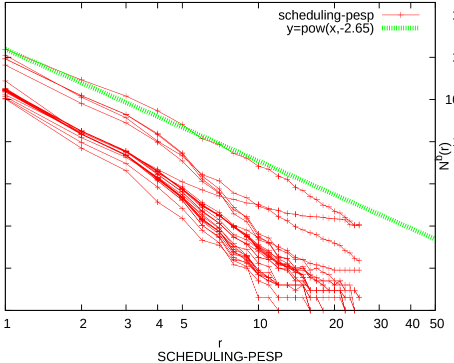
- Most families have homogeneous behaviors
- The size of the formulas does not affect the value of the dimension of the family (same slope for all instances).

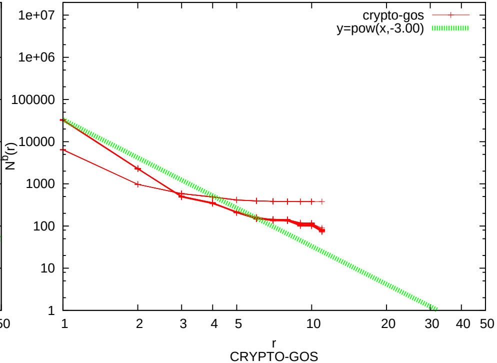
# Industrial SAT Formulas (II)




- Family *diagnosis*: $d \approx 2.84$ (26 instances)
- Family *crypto-sha*: $d \approx 3.91$ (30 instances)

# Industrial SAT Formulas (III)

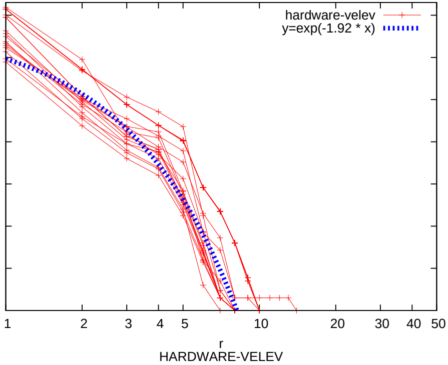



- Family *scheduling-pesp*: $d \approx 2.65$ (30 instances)
- Family *crypto-gos*: $d \approx 3.00$ (30 instances)

# Industrial SAT Formulas (IV)

- In some families, all instances have a $N(r)$ function with exponential decay, i.e. they are **not self-similar**.

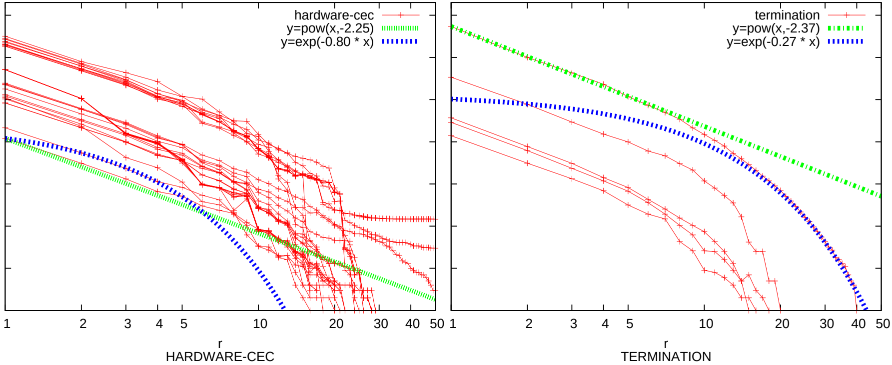


# Analyzing the Fractal Dimension (I)

We identify **two phenomena** (only in some cases):

1. **Abrupt decay** (but no exponential function).

# Analyzing the Fractal Dimension (II)




- Family *hardware-cec*: $d \approx 2.25$ (30 instances)
- Family *termination*: $d \approx 2.37$ (5 instances)

# Analyzing the Fractal Dimension (III)

We identify **two phenomena** (only in some cases):

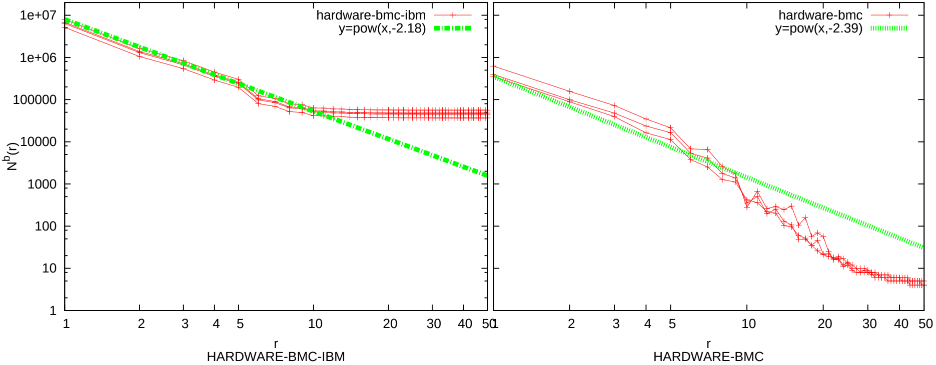
1. **Abrupt decay** (but no exponential function).
   - **Small number of edges connecting distant areas** of the graph.
     - No effect for small values of $r$.
     - They may drop down the number of circles for big values of $r$.
   - Existence of **non-*local*** dependencies.
2. **Long tail**.

# Analyzing the Fractal Dimension (III)

We identify **two phenomena** (only in some cases):

1. **Abrupt decay** (but no exponential function).
   - **Small number of edges connecting distant areas** of the graph.
     - No effect for small values of $r$.
     - They may drop down the number of circles for big values of $r$.
   - Existence of **non-*local*** dependencies.
2. **Long tail**.

# Analyzing the Fractal Dimension (IV)




- Family *hardware-bmc-ibm*: $d \approx 2.18$ (4 instances)
- Family *hardware-bmc*: $d \approx 2.29$ (3 instances)

# Analyzing the Fractal Dimension (V)

We identify **two phenomena** (only in some cases):

1. **Abrupt decay** (but no exponential function).
   - **Small number of edges connecting distant areas** of the graph.
     - No effect for small values of $r$.
     - They may drop down the number of circles for big values of $r$.
   - Existence of **non-*local*** dependencies.
2. **Long tail**.
   - Existence of (small) **unconnected components**.
   - Removed after preprocessing.

# Outline

## Summary

- **FD** related to **diameter**, but **more stable** (independent on the size).
- **BND**: efficient computation of FD in very large graphs (as SAT instances).
- Most industrial SAT instances are **self-similar**: $2 \leq d \leq 4$.
- **Random** SAT formulas are clearly **not self-similar**.
- **Learning** does not contribute to connect distant parts of the formula (as one could think) *[See details in Section 5]*.
- **Future work**: **Generators** of **more realistic industrial-like SAT instances** take into account the **fractal dimension**.

## Summary

- **FD** related to **diameter**, but **more stable** (independent on the size).
- **BND**: efficient computation of FD in very large graphs (as SAT instances).
- Most industrial SAT instances are **self-similar**: $2 \leq d \leq 4$.
- **Random** SAT formulas are clearly **not self-similar**.
- **Learning** does not contribute to connect distant parts of the formula (as one could think) *[See details in Section 5]*.
- **Future work**: **Generators** of **more realistic industrial-like SAT instances** take into account the **fractal dimension**.

## Summary

- **FD** related to **diameter**, but **more stable** (independent on the size).
- **BND**: efficient computation of FD in very large graphs (as SAT instances).
- Most industrial SAT instances are **self-similar**: $2 \leq d \leq 4$.
- **Random** SAT formulas are clearly **not self-similar**.
- **Learning** does not contribute to connect distant parts of the formula (as one could think) *[See details in Section 5]*.
- **Future work**: **Generators** of **more realistic industrial-like SAT instances** take into account the **fractal dimension**.

# Summary

- **FD** related to **diameter**, but **more stable** (independent on the size).
- **BND**: efficient computation of FD in very large graphs (as SAT instances).
- Most industrial SAT instances are **self-similar**: $2 \leq d \leq 4$.
- **Random** SAT formulas are clearly **not self-similar**.
- **Learning** does not contribute to connect distant parts of the formula (as one could think) *[See details in Section 5]*.
- **Future work**: **Generators** of **more realistic industrial-like SAT instances** take into account the **fractal dimension**.

# Thank you for your attention!

# The Fractal Dimension of SAT Formulas

C. Ansótegui[1] M.L. Bonet[2] **J. Giráldez-Cru**[3] J. Levy[3]

[1]DIEI - Univ. de Lleida

[2]LSI - UPC

[3]**IIIA-CSIC**

IJCAR 2014
July 19, 2014