Evaluation of ILP-based Approaches for Partitioning into Colorful Components

Sharon Bruckner[1] Falk Hüffner[2]
Christian Komusiewicz[2] Rolf Niedermeier[2]

[1]Institut für Mathematik, Freie Universität Berlin

[2]Institut für Softwaretechnik und Theoretische Informatik, TU Berlin

5 June 2013

Wikipedia interlanguage links

Labyrinthulomycetes – Wikipedia, the free encyclopedia – Iceweasel

File Edit View History Bookmarks Tools Help

W Labyrinthulomycetes - Wikipedi...

en.wikipedia.org/wiki/Labyrinthulomycetes

Google

Log in / create account

WIKIPEDIA
The Free Encyclopedia

Article Talk Read Edit View history Search

Labyrinthulomycetes

From Wikipedia, the free encyclopedia

The **Labyrinthulomycetes** (ICBN) or **Labyrinthulea**[1] (ICZN), or **Slime nets** are a class of protists that produce a network of filaments or tubes,[2] which serve as tracks for the cells to glide along and absorb nutrients for them. There are two main groups, the labyrinthulids and thraustochytrids. They are mostly marine, commonly found as parasites on alga and seagrass or as decomposers on dead plant material. They also include some parasites of marine invertebrates.

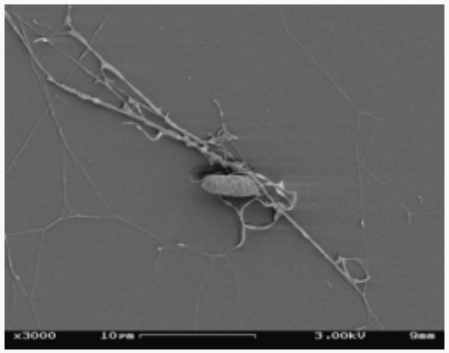
Although they are outside the cells, the filaments are surrounded by a membrane. They are formed and connected with the cytoplasm by a unique organelle called a sagenogen or bothrosome. The cells are uninucleate and typically ovoid, and move back and forth along the amorphous network at speeds varying from 5-150 µm per minute. Among the labyrinthulids the cells are enclosed within the tubes, and among the thraustochytrids they are attached to their sides.

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Toolbox
Print/export

Languages
Česky
Deutsch
Español
日本語
Македонски
Norsk (bokmål)

Slime nets



The cell with the network of filaments *Aplanochytrium sp.*

Scientific classification

Domain: Eukaryota
Kingdom: Chromalveolata
Phylum: Heterokontophyta
Class: **Labyrinthulomycetes** DICK, 2001 or

Wikipedia interlanguage links

Labyrinthulomycetes – Wikipedia, the free encyclopedia – Iceweasel

File Edit View History Bookmarks Tools Help

W Labyrinthulomycetes - Wikipedi...

en.wikipedia.org/wiki/Labyrinthulomycetes

Google

Log in / create account

WIKIPEDIA
The Free Encyclopedia

Article Talk Read Edit View history Search

Labyrinthulomycetes

From Wikipedia, the free encyclopedia

The **Labyrinthulomycetes** (ICBN) or **Labyrinthulea**[1] (ICZN), or **Slime nets** are a class of protists that produce a network of filaments or tubes,[2] which serve as tracks for the cells to glide along and absorb nutrients for them. There are two main groups, the labyrinthulids and thraustochytrids. They are mostly marine, commonly found as parasites on alga and seagrass or as decomposers on dead plant material. They also include some parasites of marine invertebrates.

Although they are outside the cells, the filaments are

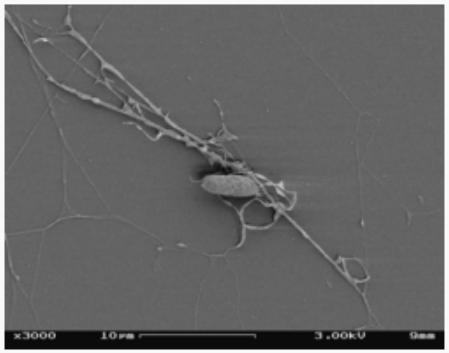
attached to their sides.

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Toolbox
Print/export

Languages
Česky
Deutsch
Español
日本語
Македонски
Norsk (bokmål)

Slime nets



The cell with the network of filaments *Aplanochytrium sp.*

Scientific classification

Domain: Eukaryota
Kingdom: Chromalveolata
Phylum: Heterokontophyta
Class: **Labyrinthulomycetes** DICK, 2001 or

Netzschleimpilze

Die **Netzschleimpilze** oder **Schleimnetze** (Labyrinthulomycetes) bilden ein Taxon innerhalb der Stramenopilen und sind somit näher mit Braunalgen, Goldalgen

Wrong interlanguage links

Schinken (German) $\rightarrow$ Prosciutto (Italian) $\rightarrow$ Пршут (Russian) $\rightarrow$ Parmaschinken (German)

Wrong interlanguage links

Schinken (German) $\rightarrow$ Prosciutto (Italian) $\rightarrow$ Пршут (Russian) $\rightarrow$ Parmaschinken (German)

Assumption

If there is a link path from a word in some language to a different word in the same language, then at least one of the links on the path is wrong.

Wrong interlanguage links

Schinken (German) $\rightarrow$ Prosciutto (Italian) $\rightarrow$ Пршут (Russian) $\rightarrow$ Parmaschinken (German)

Assumption

If there is a link path from a word in some language to a different word in the same language, then at least one of the links on the path is wrong.

Poblem

How can we fix the inconsistencies?

Model

Colorful Components

Instance: An undirected graph $G = (V, E)$ and a coloring of the vertices $\chi : V \to \{1, \ldots, c\}$.
Task: Delete a minimum number of edges such that all connected components are *colorful*, that is, they do not contain two vertices of the same color.

Applications of Colorful Components

General scenario: Record linkage

Matching entities between different databases, where links between entities are fuzzy.

- Matching items in online shop price comparison
- Matching user profiles across different social networks
- ...

Known results

- COLORFUL COMPONENTS is NP-hard already with three colors.
- With c colors and k errors to be fixed, COLORFUL COMPONENTS can be solved in $O((c-1)^k \cdot m)$ time with branch-and-bound.
- COLORFUL COMPONENTS can be approximated within a factor of $c-1$ in $O(m^2)$ time.
- Several polynomial-time preprocessing rules are known.

Method 1: Implicit Hitting Set

Hitting Set

Instance: A ground set U and a set of *circuits* $S_1, \ldots, S_n$ with $S_i \subseteq U$ for $1 \leqslant i \leqslant n$.
Task: Find a minimum-size *hitting set*, that is, a set $H \subseteq U$ with $H \cap S_i \neq \emptyset$ for all $1 \leqslant i \leqslant n$.

Method 1: Implicit Hitting Set

Hitting Set

Instance: A ground set U and a set of *circuits* $S_1, \ldots, S_n$ with $S_i \subseteq U$ for $1 \leqslant i \leqslant n$.
Task: Find a minimum-size *hitting set*, that is, a set $H \subseteq U$ with $H \cap S_i \neq \emptyset$ for all $1 \leqslant i \leqslant n$.

Observation

We can reduce Colorful Components to Hitting Set: The ground set U is the set of edges, and the circuits to be hit are the paths between identically-colored vertices.

Method 1: Implicit Hitting Set

Hitting Set

Instance: A ground set U and a set of *circuits* $S_1, \ldots, S_n$ with $S_i \subseteq U$ for $1 \leqslant i \leqslant n$.
Task: Find a minimum-size *hitting set*, that is, a set $H \subseteq U$ with $H \cap S_i \neq \emptyset$ for all $1 \leqslant i \leqslant n$.

Observation

We can reduce Colorful Components to Hitting Set: The ground set U is the set of edges, and the circuits to be hit are the paths between identically-colored vertices.

Problem

Exponentially many circuits!

Method 1: Implicit Hitting Set

v In an *implicit hitting set* problem, the circuits have an implicit description, and a polynomial-time oracle is available that, given a putative hitting set H, either confirms that H is a hitting set or produces a circuit that is not hit by H.

Method 1: Implicit Hitting Set

v In an *implicit hitting set* problem, the circuits have an implicit description, and a polynomial-time oracle is available that, given a putative hitting set H, either confirms that H is a hitting set or produces a circuit that is not hit by H.
Several approaches to solving implicit hitting set problems are known, which use an ILP solver as a black box for the HITTING SET subproblems.

Method 2: Row generation

Idea

Instead of using the ILP solver as a black box, we can use *row generation* ("*lazy constraints*"):

- Start with an empty constraint set
- When the solver finds a solution, check for a violated constraint in a callback and add it to the constraint set

Method 3: Clique Partitioning ILP formulation

Clique Partitioning

Instance: A vertex set V with a weight function $s : \binom{V}{2} \to \mathbb{Q}$.
Task: Find a cluster graph (V, E) that minimizes $\sum_{\{u,v\} \in E} s(u, v)$.

Method 3: Clique Partitioning ILP formulation

Clique Partitioning

Instance: A vertex set V with a weight function $s : \binom{V}{2} \to \mathbb{Q}$.
Task: Find a cluster graph (V, E) that minimizes $\sum_{\{u,v\} \in E} s(u, v)$.

$$s(u, v) = \begin{cases} \infty & \text{if } \chi(u) = \chi(v), \\ -1 & \text{if } \{u, v\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Method 3: Clique Partitioning ILP formulation

Clique Partitioning

Instance: A vertex set V with a weight function $s : \binom{V}{2} \to \mathbb{Q}$.
Task: Find a cluster graph (V, E) that minimizes $\sum_{\{u,v\} \in E} s(u, v)$.

$$s(u,v) = \begin{cases} \infty & \text{if } \chi(u) = \chi(v), \\ -1 & \text{if } \{u,v\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

$$e_{uv} + e_{vw} - e_{uw} \leqslant 1$$
$$e_{uv} - e_{vw} + e_{uw} \leqslant 1$$
$$-e_{uv} + e_{vw} + e_{uw} \leqslant 1$$

Cutting Planes

Definition

A *cutting plane* is a valid constraint that cuts off fractional solutions.

Cutting Planes

Definition

A *cutting plane* is a valid constraint that cuts off fractional solutions.

Tree cut

Let $T = (V_T, E_T)$ be a subgraph of G that is a tree such that all leaves L of the tree have color c, but no inner vertex has. Then

$$\sum_{uv \in E_T} (1 - e_{uv}) \geqslant |L| - 1$$

is a valid inequality.

Cutting Planes

Definition

A *cutting plane* is a valid constraint that cuts off fractional solutions.

Tree cut

Let $T = (V_T, E_T)$ be a subgraph of G that is a tree such that all leaves L of the tree have color c, but no inner vertex has. Then

$$\sum_{uv \in E_T} (1 - e_{uv}) \geqslant |L| - 1$$

is a valid inequality.

We find only tree cuts with 1 or 2 internal vertices.

Greedy Heuristics

- Merge-based:
  - Start with singleton clusters
  - Greedily merge two clusters based on cut costs and merge costs
- Move-based:
  - Start with singleton clusters
  - Greedily move one vertex from one cluster to another
  - Once no improvement is possible, merge clusters and repeat

Implementation

- Data reduction
- ILP-approaches implemented in C++ using CPLEX 12.3
- 3.4 GHz Intel Core i3-2130 with 3 MB cache and 8 GB main memory
- Source code available at `www.user.tu-berlin.de/hueffner/colcom/`

Wikipedia interlanguage links

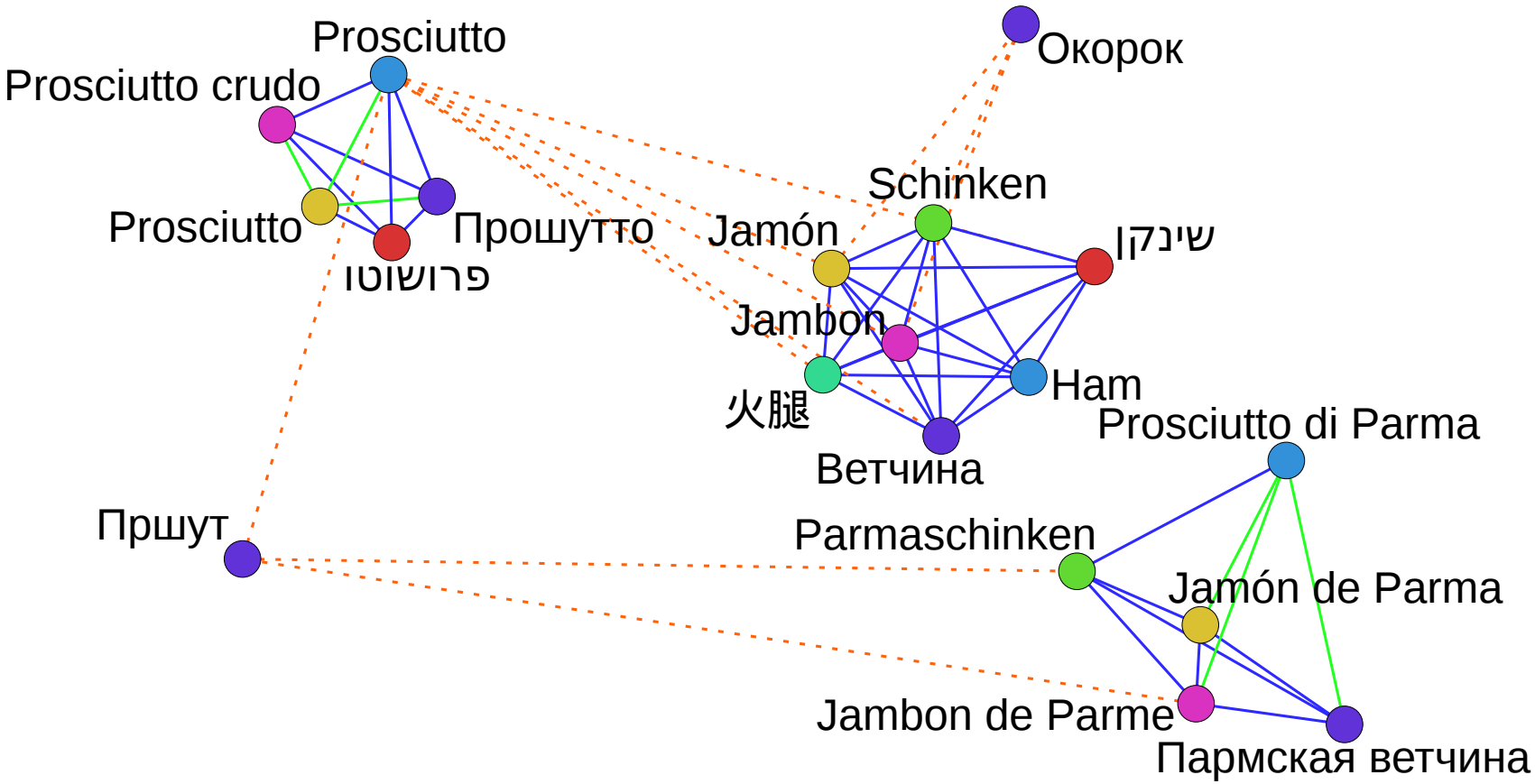
- 30 languages
- 11,977,500 vertices, 46,695,719 edges
- 2,698,241 connected components, of which 225,760 are not colorful
- largest connected component has 1,828 vertices and 14,403 edges

Wikipedia interlanguage links

- 30 languages
- 11,977,500 vertices, 46,695,719 edges
- 2,698,241 connected components, of which 225,760 are not colorful
- largest connected component has 1,828 vertices and 14,403 edges

- CLIQUE PARTITIONING algorithm finds solution in 80 minutes
- Optimal solution deletes 618,660 edges
- 434,849 suggested new links
- Merge-based heuristic has an error of 0.81 %

Wikipedia example

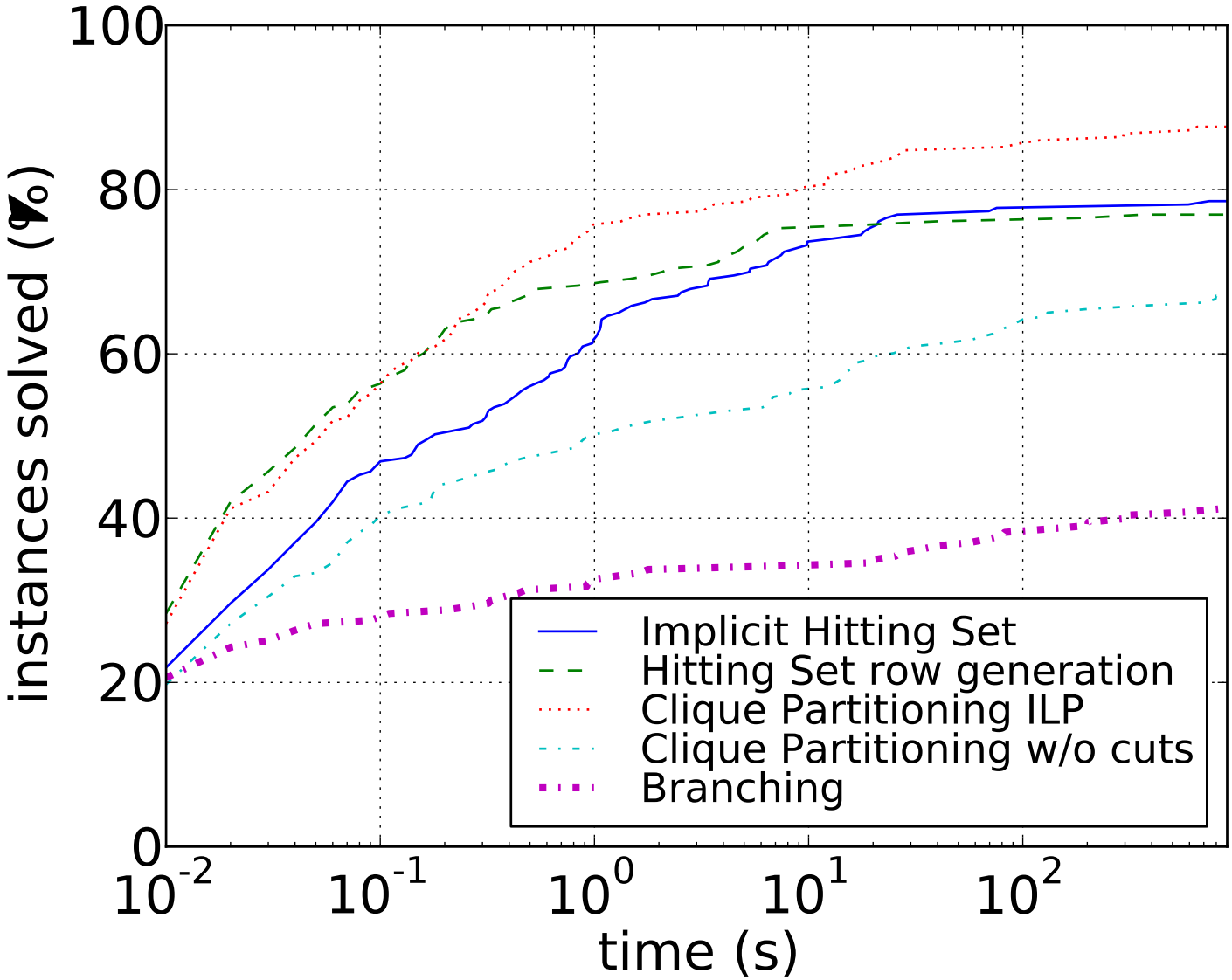


Random graph model

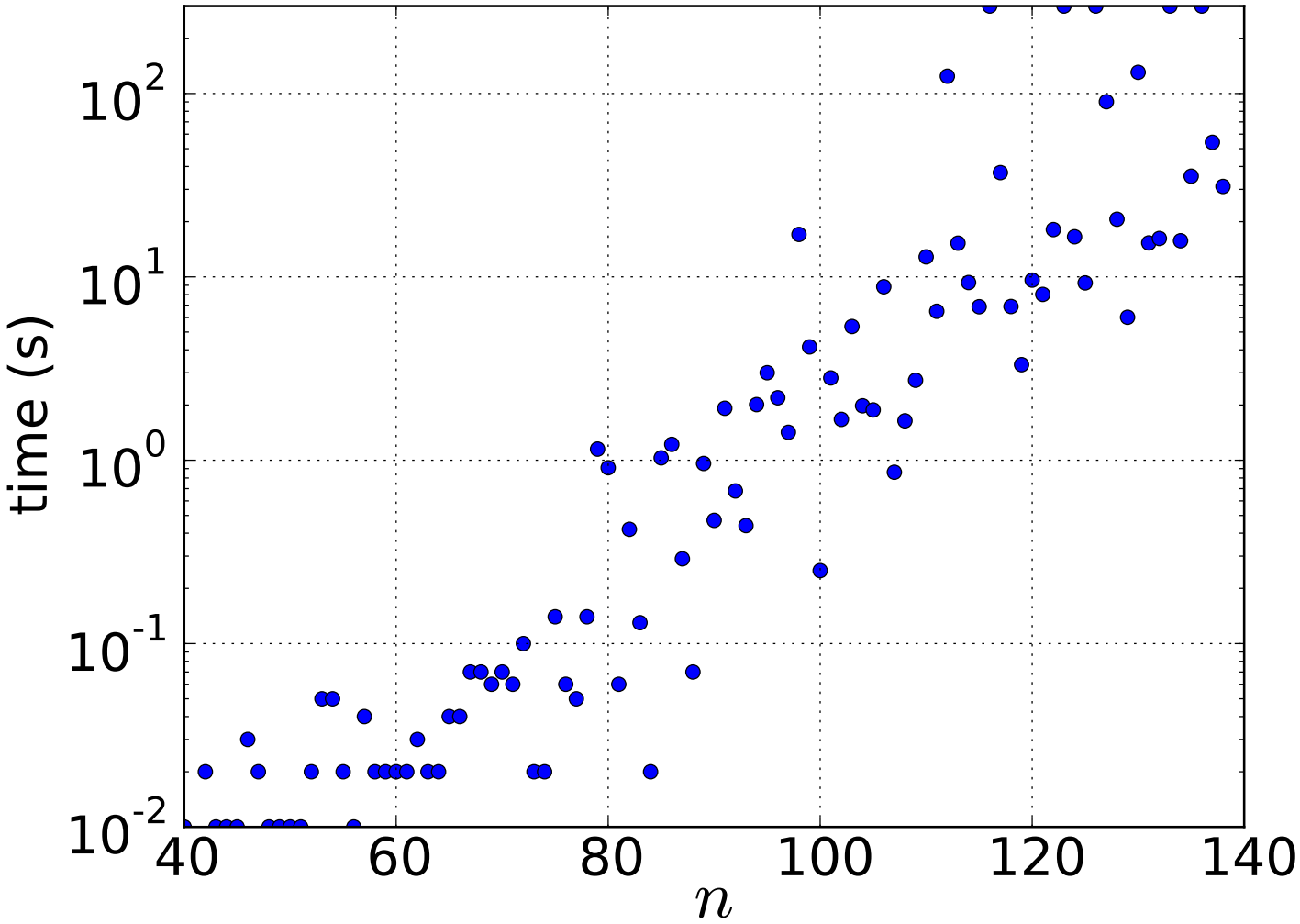
Model is the recovery of colorful components that have been perturbed.

- number of colors: $\{3, 5, 8\}$
- number of vertices: $\{60, 100, 170\}$
- probability that a component contains a vertex of a certain color: $\{0.4, 0.6, 0.9\}$
- probability that between two vertices in a component there is an edge: $\{0.4, 0.6, 0.9\}$
- probability that between two vertices from different components there is an edge: $\{0.01, 0.02, 0.04\}$.

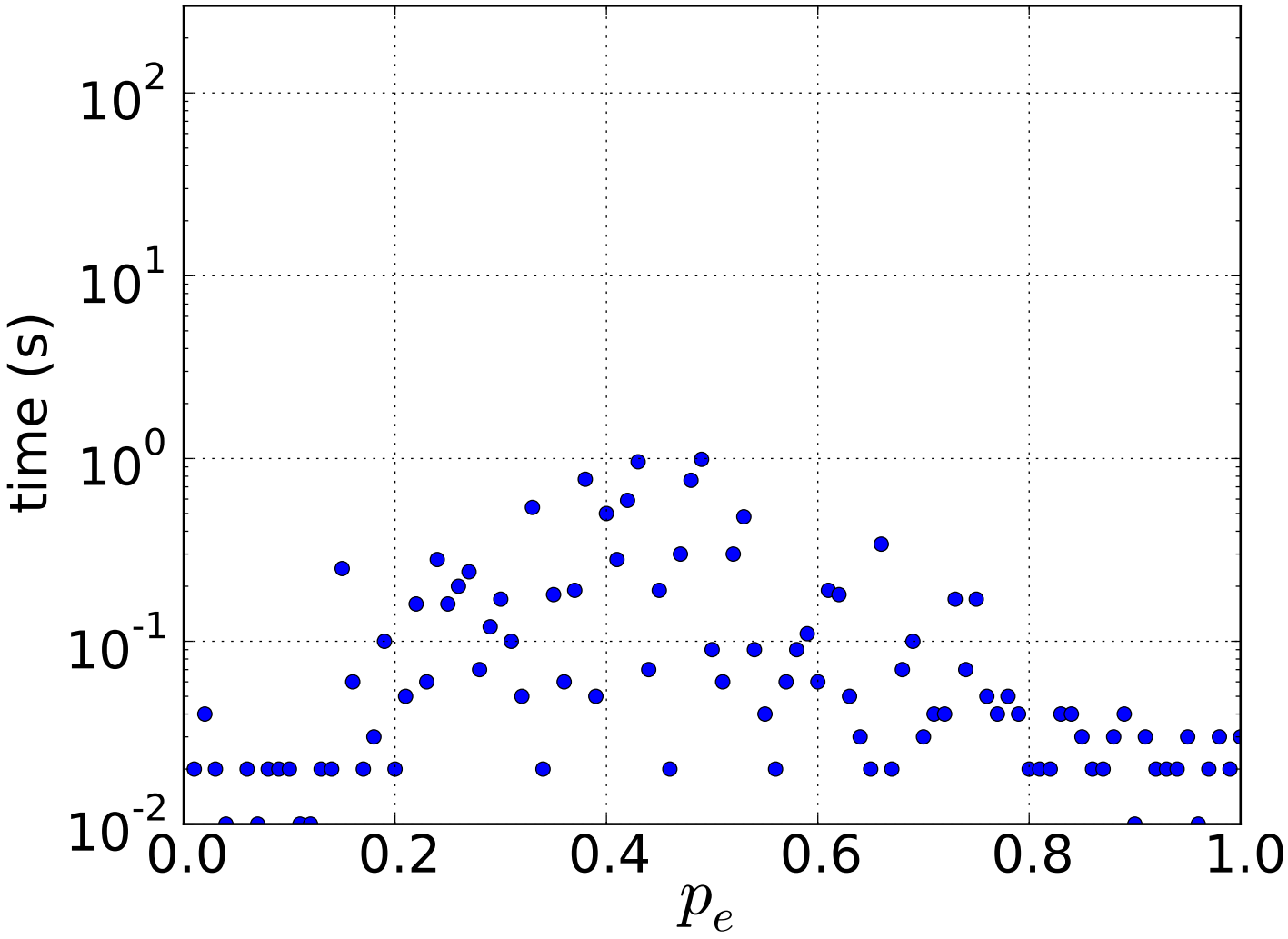
Running times for benchmark set



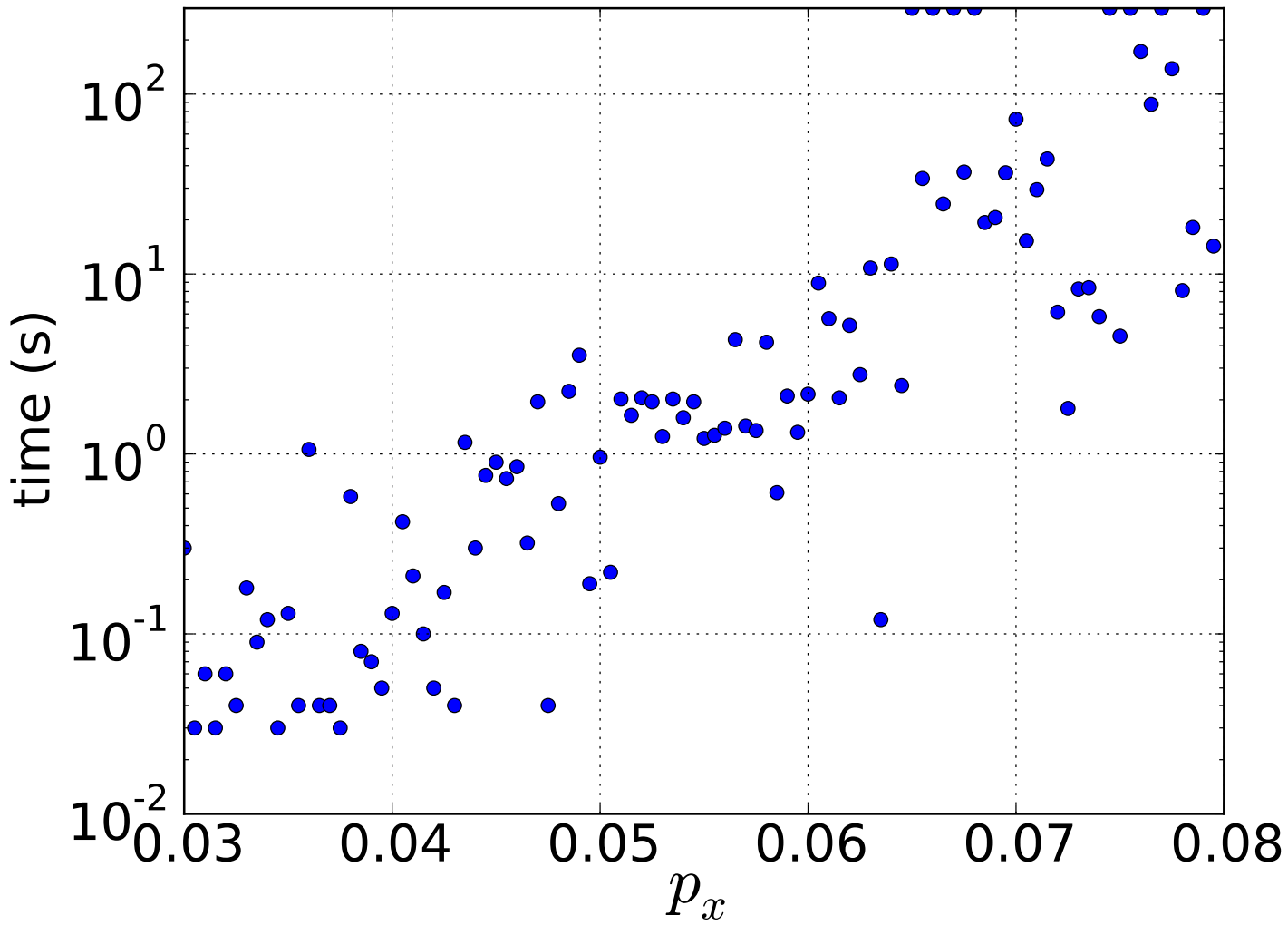
Depencency on number of vertices



Depencency on probability of intracluster edges



Depencency on probability of intercluster edges



Heuristics

Performance of the heuristics on the 213 instances where we know the optimal solution

	time	optimal	avg. error	max. error
Merge-based	$\leqslant$ 0.4 s	124	0.9 %	12.5 %
Move-based	$\leqslant$ 0.4 s	55	4.9 %	38.7 %

Outlook

Model modifications:

- more demands than just “connected” on cluster
- allows constant number of duplicates per cluster

Algorithmic improvements:

- cutting planes that take colors into account
- column generation