

# Developing H2PC for BN Structure Learning: A Comparative Study

**Lorenzo Venturi**

Master's Degree in Artificial Intelligence, University of Bologna  
lorenzo.venturi14@studio.unibo.it

April 1, 2025

## Abstract

This project aims to implement and test the H2PC algorithm, a method for structure learning in the context of Bayesian Networks. The algorithm is compared to the state-of-the-art Max-Min-Hill-Climbing (Tsamardinos, Brown, and Aliferis 2006).

The implementation is done with Python and the PyAgrum library.

The visual representation of the networks and tests are generated with PGMPY and Graphviz libraries.

The results show that H2PC is able to build a more accurate Bayesian Network structure compared to the rival; in particular this implementation shows an extremely low false-positive edge ratio and good performance even with a small fraction of the dataset available.

## Introduction

### Domain

The idea for this project comes from the paper (Gasse, Aussem, and Elghazel 2014), where the authors developed an R implementation of the algorithm and compared it with the then state-of-the-art Max-Min-Hill-Climbing algorithm.

H2PC is a hybrid method. It first learns an undirected skeleton of the structure using the HPC method (a constraint-based approach) and then performs edge orientation using a Bayesian-scoring greedy hill-climbing approach combined with a Tabu list (a search-score approach).

There are two key terms involved in the implementation:

- **Markov Blanket:** Given a target variable  $T$ , the Markov blanket  $M_T$  is any set of variables such that  $T$  is conditionally independent of all the remaining variables given  $M_T$ .
- **Markov Boundary:** The Markov boundary  $MB_T$  is the minimal Markov blanket, meaning that any subset of  $MB_T$  is no longer a Markov blanket.

HPC takes as input a target node  $T$ , a dataset  $D$ , and a set of other variables, returning an estimation of the parent-child set  $PC_T$ . The algorithm consists of three key subroutines:

- **DE\_PCS:** Data-Efficient Parents and Children Superset – Searches for a superset of  $PC_T$  with a strict restriction on conditioning size  $|Z| \leq 1$ . This improves scalability

as the number of network variables increases and helps reduce false-negative errors.

- **DE\_SPS:** Data-Efficient Spouses Superset – Searches for a superset of  $SP_T$  with a restriction on conditioning size  $|Z| \leq 2$ .
- **FDR-IAPC:** Incremental Association Parents and Children with False Discovery Rate control—An extension of the IAMB-FDR algorithm (Peña 2008). It refines the Markov Boundary by identifying the  $PC_T$  controlling the false positive edges.

### Aim

The purpose of this project is to implement and test in Python the described H2PC algorithm with various complexity and size datasets to evaluate its ability to predict the structure of the BN even in not perfect conditions.

### Method

The implementation of the project required three main phases:

1. **Algorithm Implementation:** First, the HPC algorithm was implemented, followed by H2PC. The PyAgrum library was extensively used, particularly the `BnLearner` class for computing independence tests and the SS phase (which combines Hill Climbing with a Tabu list).
2. **Network Comparison Test:** Three datasets of varying complexity were tested with different input data sizes. Using the pgmpy library, the models built with MMHC and H2PC were compared for each dataset size, using the true Bayesian network as a reference.
3. **Network Visualization and Comparison:** The Graphviz functionality in pgmpy was used to visualize the Bayesian networks, highlighting the main differences from the ground truth.

### Results

The results show that the H2PC algorithm outperforms the Max-Min algorithm as network complexity increases. Unlike the findings in the original paper, this specific implementation exhibited significantly faster computational performance compared to the pgmpy implementation of MMHC, especially as the datasets size increased.

## Analysis

## Experimental setup

To evaluate the BN, these metrics are compared between the 2 algorithms:

- **Structure Metrics** (on the skeleton of the network):
  - **False Positive Edge Ratio**
  - **Structure Precision:** number of true positive edges in divided by the total number of edges in the output.
  - **Structure Recall:** The number of true positive edges divided by the total number of true edges.

These metrics evaluate the HPC phase of H2PC against the Max-Min Parents and Children (MMPC) phase of MMHC.

- **Performance Indicators** (on the DAG):
  - **SHD (Structural Hamming Distance):** Measures the structural difference between the learned and true network as the number of add or delete an undirected edge, and add, remove, or reverse the orientation of an edge to reach the true structure.
  - **BIC Score:** Evaluates the goodness of fit of the learned model (higher is better)
  - **Computational Time Complexity:** Assesses the efficiency of the algorithm.

Three different datasets taken from the `bnlearn` repository, each presenting increasing complexity in terms of nodes and dependencies:

- **Asia Dataset:** 8 nodes, 8 arcs
- **Sachs Dataset:** 11 nodes, 17 arcs
- **Alarm Dataset:** 37 nodes, 46 arcs

Each test is executed on the same dataset varying the size to evaluate the model’s generalization ability.

## Results

Here the main results across the three datasets are showed. For the first two datasets, the ratio of H2PC metrics to MMHC metrics is compared on the DAG built on different dataset sizes. Green indicates that H2PC outperforms MMHC, red the opposite and white means no difference. The third dataset is evaluated only using H2PC, as the MMHC implementation in pgmpy exceeded the computational time limit. The final DAG for this dataset is presented below. For an complete overview of the results, please refer to the available Notebook.

Metrics	100	500	1000	2000	3000	5000	8000
FPE	0.00	0.33	0.00	0.00	0.00	1.00	0.00
Precision	2.00	1.50	1.29	1.33	1.29	1.00	1.50
Recall	1.50	1.50	0.86	1.00	0.86	0.86	1.00
SHD	0.89	0.40	0.50	0.57	0.57	0.80	0.22
BIC Score	1.018	0.995	0.991	0.992	0.995	0.998	0.987
Ex.Time	0.018	0.011	0.012	0.057	0.082	0.139	0.148

Table 1: Metrics ratio comparison on Asia Dataset

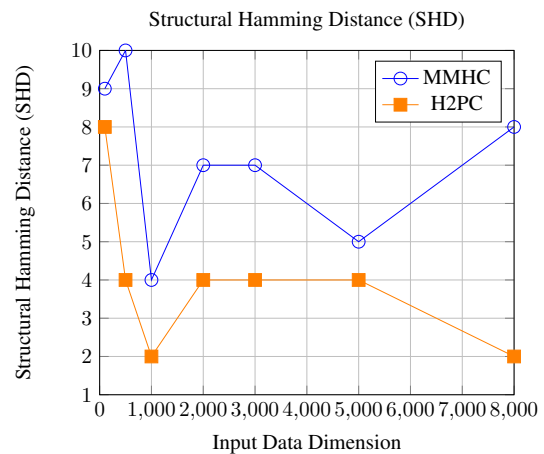


Figure 1: Comparison of Structural Hamming Distance (SHD) for MMHC and H2PC on Asia Dataset

Metrics	100	200	400	600	800
FPE	0.00	0.187	0.250	0.119	0.00
Precision	1.500	1.650	1.375	2.321	1.429
Recall	1.250	2.200	1.833	3.250	2.000
SHD	0.824	0.579	0.765	0.450	0.471
BIC Score	0.942	0.956	0.950	0.950	1.012
Ex. Time	7.9e-5	9.9e-5	1.99e-4	1.30e-4	9.7e-5

Table 2: Metrics ratio comparison on Sachs Dataset

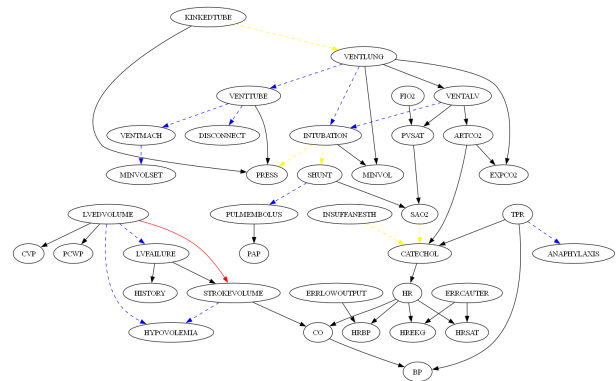


Figure 2: Final DAG of the Alarm dataset

Figure 3: Blue lines: reversed node, yellow line: missed edge, red line: wrong edge

## Conclusion

During the development of this project, i gained a deep understanding of how computational complexity increases as the network expands. Despite this, H2PC consistently outperforms MMHC. In particular, the algorithm demonstrates strong reliability in minimizing false positive edges. If combined with expert knowledge of the domain, it surely can obtain very reliable results.

## Links to external resources

- Datasets are taken from: <https://www.bnlearn.com/bnrepository/>

## References

Gasse, M.; Aussem, A.; and Elghazel, H. 2014. A hybrid algorithm for bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications* 41(15):6755–6772.

Peña, J. M. 2008. Learning gaussian graphical models of gene networks with false discovery rate control. In Marchiori, E., and Moore, J. H., eds., *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 165–176. Berlin, Heidelberg: Springer Berlin Heidelberg.

Tsamardinos, I.; Brown, L.; and Aliferis, C. 2006. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning* 65:31–78.