Inferring large graphs with an l1-penalized likelihood formulation and a hybrid genetic algorithm

Magali Champion, Victor Picheny, Matthieu Vignes

Inferring large graphs with an ℓ_1-penalized likelihood formulation and a hybrid genetic algorithm

Magali Champion
Stanford Center for Biomedical Informatics Research (BMIR),
Department of Medicine, Stanford University, CA, USA
Victor Picheny
INRA, UR875 Applied Mathematics and Computer Science Unit,
Castanet-Tolosan, France
and
Matthieu Vignes
Institute of Fundamental Sciences, Massey University,
Palmerston North, New Zealand.

Abstract

We address the issue of recovering the structure of large sparse directed acyclic graphs from noisy observations of the system. We propose a novel procedure based on a specific formulation of the ℓ_1-norm regularized maximum likelihood, which decomposes the graph estimation into two sub-problems: topological structure and node order learning. We provide oracle inequalities for the graph estimator, as well as an algorithm to solve the induced optimization problem, in the form of a convex program embedded in a genetic algorithm. We apply our method to various data sets (including data from the DREAM4 challenge) and show that it compares favorably to state-of-the-art methods.

Keywords: Directed Acyclic Graphs, Lasso, Convex program, Optimization.

1 Introduction

Revealing the true structure of a complex system is paramount in many fields to identify system regulators, predict its behavior or decide where interventions are needed to disentangle direct relationships (Newman, 2003; Barabási & Oltvai, 2004; Souma et al., 2006; Verma et al., 2014). This problem can often be seen as a graph inference problem: given observational data, we aim at predicting the presence (or absence) of edges between elements, which form the vertices of a graph. As a motivating problem, the reconstruction of Gene Regulatory Networks (GRN), which model activation and inhibition relationships between genes, is one of the main challenges in modern computational biology.

A popular approach consists in assuming that the data are generated by a Directed Acyclic Graph (DAG) (Pearl, 2009). DAGs are made of a collection of vertices, which stand for variables, and directed edges to model the dependency structure among the variables, avoiding loops and cycles. However, inferring a DAG is a rather challenging problem. Firstly, the number of nodes p of the graph may be so large that exploring relevant DAG topologies is simply infeasible, since the number of possible DAG structures is super-exponential in p (Koivisto & Sood, 2004; Tsarmadinos et al., 2006; Grzegorczyk & Husmeier, 2008). Another dimension flaw occurs when p, even being reasonable, is larger than the number of observations, and model/parameter estimation is jeopardized. High-dimensional statistical techniques are then needed to overcome this issue (Bühlmann & van de Geer, 2011; Giraud, 2014). Secondly, even if the ratio between p and the sample size n is not impeding model estimation, the nature of the data can be an additional obstacle (Ellis & Wong, 2008; Guyon et al., 2010; Fu & Zhou, 2013). The available observational data are in general not sufficient to identify the true underlying DAG, and can only determine an equivalence class of DAGs (Verma & Pearl, 1991). This approach relies on the assumption that the joint distribution is Markov and faithful with respect to the true graph (Spirtes et al., 2000).

A large number of methods have been proposed for estimating DAGs, including for instance score-based methods (Bayesian score, Friedman & Koller 2003 or Bayesian Information Criterion, Schwarz 1978), penalized likelihood methods (Shojaie & Michailidis, 2010), complex space sampling (Zhou, 2011) or the PC algorithm (Kalisch & Bühlmann, 2007). The latter has been proved to be uniformly consistant in the high-dimensional case, but requires a test of conditional independences that quickly becomes computationally intractable.

In this work, we focus on Gaussian structural equation models (Pearl, 2009) with equal noise variances, for which the identifiability of the whole DAG is satisfied (Peters et al., 2014), associated with maximum likelihood estimators (MLE). In the last years, the ℓ_0-regularization of the MLE has been the focus of a large number of works since it leads to infer sparse graphs. For a known order among the variables in the graph, Shojaie & Michailidis (2010) present results for the estimation of high-dimensional graphs based on independent linear regressions using an adaptive lasso scheme. When the order of the variables is unknown, van de Geer & Bühlmann (2013) studied the convergence of the ℓ_0-penalized likelihood. However, from a computational point of view, the ℓ_0-regularized approaches (Silander & Myllymäki, 2006; Hauser & Bühlmann, 2012) require an exhaustive exploration of the set of DAGs, impractical for estimating graphs with more than 20 vertices.

Our objective is to overcome this drastic dimensional limitation and find inference

strategies for graphs with up to several hundred nodes. Such strategies must ensure a high level of sparsity and be supported by computationally affordable algorithms, while preserving sound theoretical bases. Here, we propose to use the ℓ_1-regularization, similarly to Fu & Zhou (2013) and Shojaie & Michailidis (2010), to define the MLE. From a computational point of view, this regularization makes the criterion to maximize partially convex while ensuring sparse estimates. Our contribution is two-fold: first, we provide oracle inequalities that guarantee good theoretical performances of our proposed estimator in the sparse high-dimensional setting; then, we provide an efficient algorithm to infer the true unknown DAG, in the form of a convex program embedded in a genetic algorithm.

The next section covers the model definition and the associated penalized MLE problem. Section 3 details the oracle inequalities, and Section 4 our inference algorithm. Section 5 reports numerical experiments both on toy problems and realistic data sets.

2 The ℓ_1-penalized likelihood for estimating DAGs

2.1 DAG's modelling and estimation

In this work, we consider the framework of an unknown DAG $\mathcal{G}_0 = (V, E)$, consisting of vertices $V = \{1, ..., p\}$ and a set of edges $E \subseteq V \times V$. The p nodes are associated to random variables $X^1, ..., X^p$. A natural approach, developped by Meinshausen & Bühlmann (2006) to solve the network inference problem is to consider that each variable X^i $(1 \leq i \leq p)$ of the DAG can be represented as a linear function of all other variables X^j $(j \neq i)$ through the Gaussian Structural Equation Model:

$$\forall j \in [\![1, p]\!], \quad X^j = \sum_{i=1}^{p} (G_0)_i^j X^i + \varepsilon^j, \tag{1}$$

with $\varepsilon^j \sim \mathcal{N}(0, \sigma^2)$ (σ^2 known, independent of j) a Gaussian residual error term and E corresponding to the non-zero coefficients of G_0, *i.e.* $(G_0)_i^j$ encoding the relationship from variable X^i to variable X^j.

Assume that we observe an n-sample consisting of n i.i.d. realizations $(X^1, ..., X^p)$ from Equation (1). We denote by $X := (X^1, ..., X^p)$ the $n \times p$ data matrix with n i.i.d. rows, distributed according to a $\mathcal{N}(0, \Sigma)$ law, where Σ is a non-singular covariance matrix. The relations between the variables can be represented in its matrix form:

$$X = XG_0 + \varepsilon, \tag{2}$$

where $G_0 = ((G_0)_i^j)_{1 \leq i,j \leq p}$ is the $p \times p$ matrix compatible with the graph $\mathcal{G}_0$ and $\varepsilon := (\varepsilon^1, ..., \varepsilon^p)$ is the $n \times p$ matrix of noise vectors.

The negative log-likelihood of the model is then (Rau et al., 2013):

$$\ell(G) = \frac{np}{2} \log(2\pi) + n \log \sigma + \frac{1}{\sigma^2} \sum_{k=1}^{n} \sum_{j=1}^{p} \left(X_k (I - G)^j \right)^2. \tag{3}$$

To recover the structure of the DAG $\mathcal{G}_0$ and make the estimated graph sparse enough, we focus on a penalized maximum likelihood procedure (Bickel & Li, 2006):

$$\hat{G} = \underset{G \in \mathcal{G}_{DAG}}{\operatorname{argmin}} \ \{\ell(G) + \lambda \operatorname{pen}(G)\}, \tag{4}$$

where $\ell(.)$ is the negative log-likelihood of Equation (3), pen(.) is a determined penalization function, λ is a trade-off parameter between penalization and fit to the data, and $\mathcal{G}_{DAG}$ is the set of $p \times p$ matrices compatible with a DAG over p nodes.

Using an ℓ_0-norm regularization in Equation (4) to infer sparse graphs is an attractive option, since the criterion to minimize is constant for all equivalent DAGs. It guarantees that we can recover the Markov equivalence class of the underlying DAG. From a computational point of view, the main difficulty when solving Equation (4) is to explore $\mathcal{G}_{DAG}$, which is a well-known NP-hard problem (Chickering, 1996): an ℓ_0-regularization does not set a favorable framework for this task. To avoid the whole exploration of $\mathcal{G}_{DAG}$, a dynamic programming method has been proposed in Silander & Myllymäki (2006), using a particular decomposition of the ℓ_0-penalized maximum likelihood. The greedy equivalent search algorithms of Chickering (2002); Hauser & Bühlmann (2012) restrict the search space to the smaller space of equivalence classes and provide an efficient algorithm without enumerating all the equivalent DAGs. They were shown to be asymptotically optimal under a faithfulness assumption (i.e. independence in the distribution are those read from $\mathcal{G}_0$). However, these approaches cannot be used on high-dimensional data to estimate graphs with a large number of nodes.

We consider the setting of Gaussian structural equation model with equal noise variances. Peters et al. (2011, 2014) showed that the true DAG is identifiable for respectively discrete and continuous data. We focus on the ℓ_1-norm convex regularization instead of ℓ_0 for its sparse, high-dimensional and computational properties. This regularization clearly improves the computation in Equation (4) with a convex constraint on the graph topology.

Given Equation (3) and omitting constant terms, the ℓ_1-penalized likelihood estimator we consider is:

$$\hat{G} = \underset{G \in \mathcal{G}_{DAG}}{\operatorname{argmin}} \quad \left\{ \frac{1}{n} \left\| X(I - G) \right\|_F^2 + \lambda \left\| G \right\|_1 \right\}, \tag{5}$$

where $\|.\|_1$ denotes the ℓ_1-norm and $\|.\|_F$ the Frobenius norm, i.e. respectively for any matrix $M := (M_i^j)_{1 \leq i,j \leq p}$, $\|M\|_1 = \sum_{i,j} \left| M_i^j \right|$ and $\|M\|_F = \sum_{i,j} \left(M_i^j \right)^2$.

Remark 1. *The price to pay for the ℓ_1-norm relaxation is a bias, which can be controlled by thresholding the estimator (van de Geer et al., 2011), i.e. setting to 0 small values of $\hat{G}$.*

2.2 A new formulation for the estimator

We propose here a new formulation of the minimization problem of Equation (5). It will allow us to naturally uncouple two steps of the minimisation procedure: node ordering and graph topology search. A key property is that any DAG leads to a topological ordering of its vertices, denoted $\leq$, where a potential directed edge from node X^i to node X^j is equivalent to $X^j \leq X^i$ (Kahn, 1962; Cormen et al., 2001). This ordering is not unique in general, except when there exists a directed path between all the nodes of the graph (see Example 1 below for more explanations). Proposition 2.1 from Bühlmann (2013) then gives an equivalent condition for a matrix to be compatible with a DAG.

Proposition 2.1 (Bühlmann 2013)**.** *A matrix G is compatible with a DAG $\mathcal{G}$ if and only if there exists a permutation matrix P and a strictly lower triangular matrix T such that:*

$$G = PTP^T.$$

Graphically, the permutation matrix sets an ordering of the nodes of the graph and is associated to a complete graph. The strictly lower triangular matrix T sets the graph structure, *i.e.* the non-zero entries of G, as illustrated in Example 1.

Example 1. *Consider the DAG $\mathcal{G}$ given in Figure 1 (left). The corresponding matrix G can be written as the strictly lower-triangular matrix T by permutation of its rows and columns using P:*

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 6 & 4 \\ 0 & 0 & 0 & 7 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix} = PTP^T, \text{ with } T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 7 & 0 & 0 \\ 2 & 4 & 6 & 1 & 0 \end{pmatrix} \text{ and } P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Looking at the non-zero values of P column by column, P defines a node hierarchy $X^1 \leq X^5 \leq X^4 \leq X^3 \leq X^2$ compatible with the topological orderings of $\mathcal{G}$. Graphically, P is associated to the complete graph represented in Figure 1 (right). The dashed edges then correspond to the lower zero entries of T. Note that since X^4 is not connected with X^5 and X^1, three topological ordering are possible ($X^4 \leq X^1 \leq X^5$, $X^1 \leq X^4 \leq X^5$ and $X^1 \leq X^5 \leq X^4$).

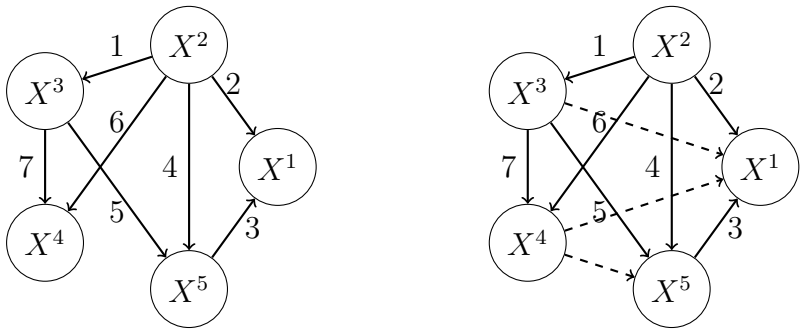


Figure 1: An example of DAG $\mathcal{G}$ (left) and the action of P and T on $\mathcal{G}$: P is associated to a complete graph that orders the nodes of the graph (right) and T sets the weights on the edges. The dashed edges correspond to null weight edges (a zero entry in T).

Using Proposition 2.1, the estimator in (5) leads to the following equivalent optimization problem:

$$(\hat{P}, \hat{T}) = \underset{(P,T)\in\mathcal{C}}{\operatorname{argmin}} \left\{ \frac{1}{n} \left\| X(I - PTP^T) \right\|_F^2 + \lambda \|T\|_1 \right\}, \tag{6}$$

where the optimization space $\mathcal{C}$ is defined as $\mathcal{C} = \mathbb{P}_p(\mathbb{R}) \times \mathbb{T}_p(\mathbb{R})$, with $\mathbb{P}_p(\mathbb{R})$ the set of permutation matrices and $\mathbb{T}_p(\mathbb{R})$ the set of strictly lower-triangular matrices. This new parametrization is particularly useful to separate the DAG structure search in two tasks: the ordering estimation and the graph structure learning.

Note that a similar formulation has already been proposed by van de Geer & Bühlmann (2013) to ensure good theoretical properties for the ℓ_0-penalized log-likelihood estimation. However, it has never been exploited from a computational point of view to recover the graph structure optimizing problem (5). In the following two sections, we propose a theoretical analysis of the proposed estimator and a computationally effficient algorithm to solve (6)

3 Oracle inequalities for the DAG estimation

The main result in this section is about convergence rates: in Theorem 1, we provide upper bound for error associated with the ℓ_1-penalized maximum likelihood estimator considered in Equation (6), both in prediction (Equation 7) and estimation (Equation 8). Following the works of van de Geer & Bühlmann (2013) on the ℓ_0-penalized maximum likelihood estimator and of Bickel et al. (2009) on the Lasso and the Dantzig Selector, we obtain two convergence results under some mild sparsity assumptions, when the number of variables is large but upper bounded by a function $\varphi(n)$ of the sample size n.

3.1 Estimating the true order of variables

For a known ordering among the variables of the graph (Shojaie & Michailidis, 2010), which is an unrealistic assumption in many applications, the DAG inference problem is rather simple. To provide oracle inequalities of the proposed estimator, in the case of unknown order we consider here, we first focus on the problem of estimating the true variable order. Let us denote by Π_0 the set of permutation matrices compatible with the true DAG $\mathcal{G}_0$:

$$\Pi_0 = \left\{ P \in \mathbb{P}_p(\mathbb{R}), P^T G_0 P \in \mathbb{T}_p(\mathbb{R}) \right\}.$$

Π_0 contains one or more permutation matrice(s) (see Example 1). We will have to make a decision as to whether the estimated order of variables $\hat{P}$ given by Equation (6) is in Π_0 or not.

To answer this question, we investigate the effect of learning an erroneous order of variables $P \notin \Pi_0$. We introduce the following notations: for any permutation matrix $P \in \mathbb{P}_p(\mathbb{R})$, we denote by $G_0(P)$ the matrix defined as:

$$G_0(P) = P T_0 P^T, \quad \text{with } T_0 = P_0^T G_0 P_0 \text{ a lower triangular decomposition of } G_0.$$

From a graphical point of view, while $P \notin \Pi_0$, the graph $\mathcal{G}_0(P)$ associated to $G_0(P)$ is obtained from $\mathcal{G}_0$ by permuting some of its nodes (see Example 2), otherwise $\mathcal{G}_0(P) = \mathcal{G}_0$. We also denote by $\varepsilon(P) := X - XG_0(P)$ the associated residual term. To simplify the theoretical results and proofs, until the end of this work, we assume that the noise variances $\sigma^2 := \mathrm{Var}(\varepsilon^j)$ are equal to one. Our results are still valid even if $\sigma^2 \neq 1$, by small modifications in the constant terms as long as they are all equal. We denote by $\Omega(P)$ the covariance matrix of $\varepsilon(P)$ and $\omega_j(P) := \mathrm{Var}(\varepsilon^j(P))$ the associated noise variances.

With these notations and checking that the assumptions presented in Section 3.2 hold, we ensure that, with large probability, we choose a right order of variables and the estimated graph converges to the true graph when n and p grow to infinity (see Section 3.3).

Example 2. *Let* $P = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \notin \Pi_0$ *a wrong permutation.*

In Figure 2, we represent the permuted graph $\mathcal{G}_0(P)$ (right) associated to the graph $\mathcal{G}_0$ (left). The latter is obtained from $\mathcal{G}_0$ after permutation of its nodes using PP_0^T, where P_0 (corresponding to the matrix P in Example 1) defines a right order of variables.

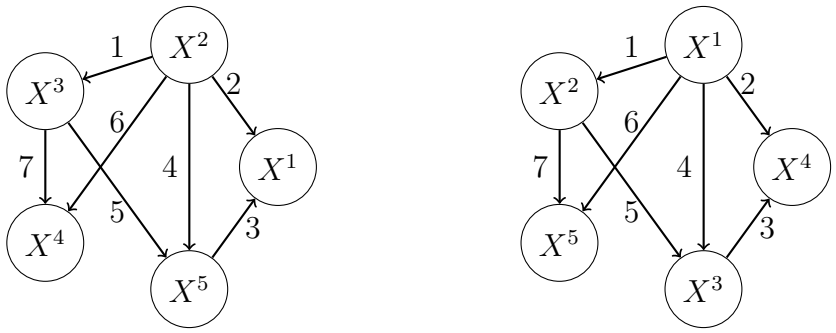


Figure 2: The graph $\mathcal{G}_0$ (left) and the permuted graph $\mathcal{G}_0(P)$ (right) associated to the permutation P.

3.2 Assumptions on the model

For a matrix $M \in \mathcal{M}_{p\times p}(\mathbb{R})$ and a subset $\mathcal{S}$ of $[\![1,p]\!]^2$, we denote by $M_{\mathcal{S}}$ the matrix that has the same elements as M on $\mathcal{S}$ and zero on the complementary set $\mathcal{S}^C$ of $\mathcal{S}$. We now introduce the assumptions we used to obtain statistical properties of our estimator.

Hypothesis

$\mathbf{H_1}$ There exists σ_0^2, independent of p and n, such that $\max_{1\leq j\leq p} \mathrm{Var}(X^j) \leq \sigma_0^2$.

$\mathbf{H_2}$ The maximal weight of the DAG $\mathcal{G}_0$ is bounded $\|G_0\|_\infty := \max_{1\leq i,j\leq p} \left|(G_0)_i^j\right| < +\infty$.

$\mathbf{H_3}$ The maximal number of parents s_{max} of the graph nodes is bounded by $s^* > 0$ depending only on $\|G_0\|_\infty$.

$\mathbf{H_{dim}}$ The number of nodes p satisfies $p^3 \log p = \mathcal{O}(n)$.

$\mathbf{H_{RE}}(s)$ There exists $\kappa(s) > 0$ with $1 \leq s \leq p^2$ such that:

$$\min\left\{ \frac{\|XM\|_F}{\sqrt{n}\,\|M_{\mathcal{S}}\|_F} : \mathcal{S} \subset [\![1,p]\!]^2, |\mathcal{S}| \leq s, M \in \mathcal{M}_{p\times p}(\mathbb{R})\backslash\{0\}, \|M_{\mathcal{S}^C}\|_1 \leq 3\,\|M_{\mathcal{S}}\|_1 \right\} \geq \kappa(s).$$

$\mathbf{H_{id}}$ There exists $0 < \eta \leq C\frac{n}{p\log p}$, with $C > 0$, such that, for all permutations $P \notin \Pi_0$,

$$\frac{1}{p}\sum_{j=1}^{p} \left(|\omega_j(P)|^2 - 1\right)^2 > \frac{1}{\eta}.$$

Assumption $\mathbf{H_1}$ is clearly satisfied for $\sigma_0^2 = 1$ if we standardize the data. Assumptions $\mathbf{H_{2-3}}$ are needed to show that the minimal eigenvalue λ_{min} of the covariance matrix Σ of X is not too small when n and p grow to infinity. It is clearly related to condition 3.2 of van de Geer & Bühlmann (2013). We relax however the latter allowing λ_{min} to decrease with n and p (see Section B of the Appendix for further details).

Assumption $\mathbf{H_{dim}}$ deserves a special attention since it strongly bounds the high dimensional setting. The considered problem is obviously non-trivial and requires a sufficient amount of information. This assumption has to be carefully compared with the beta-min

condition introduced by van de Geer & Bühlmann (2013) for the ℓ_0-regularized MLE, satisfied in a less restrictive regime $p = \mathcal{O}(\sqrt{n/\log n})$. More precisely, $\mathbf{H_{dim}}$ can be relaxed to the high-dimensional case at the expense of a tighter restriction on the maximal degree s_{max} of the graph. Note however that universal conditions cannot be overcome and the ultra-high dimension settings (e.g. Wainwright (2009); Verzelen (2012)) is an insurmountable limit, specifically when $s_{max}\log(p/s_{max})$ becomes large as compared to n.

Assumption $\mathbf{H_{RE}}(s)$ is a natural extension of the Restricted Eigenvalue condition of Bickel et al. (2009) to our multi-task setting. More precisely, denoting

$$\tilde{X} = \underbrace{\begin{pmatrix} X & & 0 \\ & \ddots & \\ 0 & & X \end{pmatrix}}_{p^2} \Bigg\updownarrow n \times p,$$

$\mathbf{H_{RE}}(s)$ is equivalent to assuming that the Gram matrix $\frac{\tilde{X}\tilde{X}^T}{n}$ is non-degenerate on a restricted cone (Lounici et al., 2009; Bühlmann & van de Geer, 2011). Notice that this condition is very classical in the literature. It yields good practical performance even for small sample sizes, and there is some hope that accurate population eigenvalues could be estimated even in a large dimension setting (Mestre, 2008; El Karoui, 2008; Liu et al., 2014; Ledoit & Wolf, 2015).

The last assumption $\mathbf{H_{id}}$ is an identifiability condition needed to ensure that the estimated permutation $\hat{P}$ is in Π_0. This assumption was introduced by van de Geer & Bühlmann (2013) as the "omega-min" condition. In a sense, it separates the set of compatible permutations from its complement in a finite sample scenario.

3.3 Main result

The result we establish in this section is double-edged: (a) with large probability, the first part of Theorem 1 ensures that the estimated $\hat{P}$ belongs to Π_0, and (b) we provide oracle inequalities both in prediction and estimation for the graph estimated from the minimisation problem (6). This result clearly states the desirable theoretical properties of the derived estimator, assuming reasonable conditions on the complex system embedding the data.

Theorem 1. *Assume that Assumptions* $\mathbf{H_{1,2,3}}$, $\mathbf{H_{dim}}$, $\mathbf{H_{RE}}(s)$ *with* $s \subset [\![1,p]\!]^2$ *such that* $\sum_{i,j} \mathbb{1}_{(G_0)_i^j \neq 0} \leq s$ *and* $\mathbf{H_{id}}$ *are satisfied. Let* $\lambda = 2C\sqrt{\frac{\log p}{n} s_{max}}$. *Then, with probability greater than* $1 - 5/p$, *any solution* $\hat{G} = \hat{P}\hat{T}\hat{P}^T$ *of the minimization problem (6) satisfies that* $\hat{P} \in \Pi_0$. *Moreover, with at least the same probability, the following inequalities hold:*

$$\frac{1}{n}\left\|X\hat{G} - XG_0\right\|_F^2 \leq \frac{16C^2 s_{max}^2 \log p}{\kappa^2(s) n}. \tag{7}$$

$$\left\|\hat{G} - G_0\right\|_1 \leq \frac{16C}{\kappa^2(s)} s_{max}^{3/2}\sqrt{\frac{\log p}{n}}. \tag{8}$$

The proof of this result is deferred in Section B of the Appendix.

Remark 2. *Theorem 1 states that with probability at least* $1 - 5/p$*, we choose a compatible order of variables over the set of permutations. Inequalities (7) and (8) give non-asymptotic upper bounds on the loss under conditions depending on* p *and* n*, the graph structure and the data.*

Remark 3. *Inequalities (7) and (8) show that the estimated* $\hat{T}$ *is close to the true* T_0 *with large probability as* $p, n \to +\infty$*.*

4 Inference algorithm

4.1 Global algorithm overview

In this section, we propose a computational procedure devoted to solve Equation (6). Although decomposing the original problem made it much easier to handle, this problem is indeed a very challenging task from an optimization point of view, due to the different nature of the variables P and T, the non-convexity of the cost function and the very high dimension of the search space.

An intuitive approach would consist in using an alternating minimization: alternatively, fix one of the variables P or T and optimize over the other one, then reverse the roles or P and T and do it again iteratively until convergence for some criterion (Csiszár & Tusnády, 1984). However, the structure of our problem does not allow us to use such a scheme: looking for an optimal T given a fixed P makes sense, but changing P for a fixed T does not.

In our inference algorithm, an outer loop is used to perform the global search among the DAGs space, which is driven by the choice of P, while a nested loop is used to find an optimal T for each given fixed P (see Figure 3). As we show in the following, population-based metaheuristics algorithms are a natural and efficient choice for exploring the space of permutation matrices (Section 4.3). The nested optimization problem can be resolved using a steepest descent approach (Section 4.2).

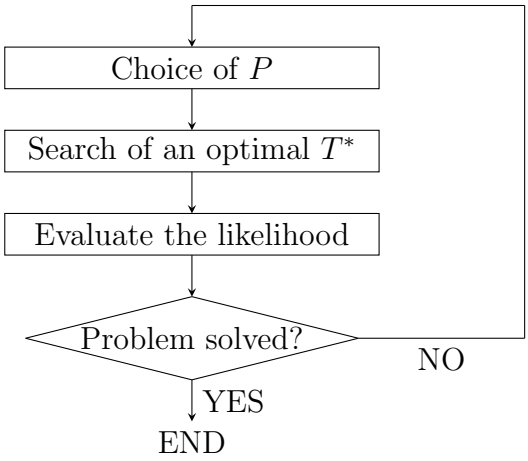


Figure 3: Overview of our hybrid algorithm.

4.2 Graph structure learning when the variable order is fixed

Assume first that the variable ordering $P \in \mathbb{P}_p(\mathbb{R})$ is fixed. The problem of inferring a graph is then reduced to estimating the graph structure, which can be solved by finding a solution of:

$$\min_{T \in \mathbb{T}_p(\mathbb{R})} \left\{ \frac{1}{n} \left\| X(I - PTP^T) \right\|_F^2 + \lambda \|T\|_1 \right\}. \tag{9}$$

Equation (9) is a well-studied problem in machine learning, as it is closely related to the ℓ_1-constrained quadratic program, known as the Lasso in the statistics literature (Tibshirani, 1996). Indeed, the ℓ_1-regularization leads to variable selection and convex constraints that make the optimization problem easy to solve. We note here that this allows us to always provide a locally optimal solution, i.e optimal weight estimates given a hierarchy between the nodes.

A large number of efficient algorithms are available for computing the entire path of solutions as λ is varied, e.g. the LARS algorithm of Efron et al. (2004) and its derivative. For example, in the context of the estimation of sparse undirected graphical models, Meinshausen & Bühlmann (2006) fit a Lasso model to each variable, using the others as predictors. The graphical Lasso (or glasso, Friedman et al. 2007) algorithm directly relies on the estimation of the inverse of a structure covariance matrix assumed to be sparse. Improvements were proposed for example by Duchi et al. (2008) (improved stopping criterion) and Witten et al. (2011) (estimation of a block-diagonal matrix). Other authors propose to solve the optimization problem using an adaptation of classical optimization methods, such as interior point (Yuan & Lin, 2007) or block coordinate descent methods (Banerjee et al., 2008; Friedman et al., 2007).

We propose here an original convex optimization algorithm to find the solution in Equation (9) in a form similar to a steepest descent algorithm. Our proposed algorithm is much quicker than a glasso approach, a desirable features as it will run at each iteration of the global algorithm (see the "Search of an optimal T^*" box in Figure 3 and the "Evaluate the new individuals" item in Algorithm 2). Moreover, its mechanistic components (see Section A of the Appendix) allowed us to derive the theoretical results of Theorem 1. The proposed scheme can be seen as an adaptation of the LARS algorithm with matrix arguments. Let $(T_k)_{k \geq 0}$ the sequence of matrices defined as:

$$\forall i, j \in [\![1, p]\!]^2, \quad (T_{k+1})_i^j = \text{sign}\left((U_k)_i^j\right) \max\left(0, \left|(U_k)_i^j\right| - \frac{\lambda}{L}\right), \tag{10}$$

where $U_k = T_k - \frac{\nabla\left(\frac{1}{n}\left\|X(I-PT_kP^T)\right\|_F^2\right)}{L}$, L is the Lipschitz constant of the gradient function $\nabla\left(\frac{1}{n}\left\|X(I-PT_kP^T)\right\|_F^2\right)$ and sign is the sign of any element. Then, a solution of (9) is given by performing Algorithm 1, where:

- the projection $\text{Proj}_{\mathbb{T}_p(\mathbb{R})}(T)$ of any $p \times p$ real-valued matrix $T = ((T_k)_i^j)_{i,j}$ on the set $\mathbb{T}_p(\mathbb{R})$ is given by

$$\left(\text{Proj}_{\mathbb{T}_p(\mathbb{R})}(T_k)\right)_i^j = \begin{cases} 0 & \text{if } i < j, \\ (T_k)_i^j & \text{otherwise.} \end{cases} \tag{11}$$

- the gradient of $\frac{1}{n}\left\|X(I - PT_kP^T)\right\|_F^2$ is

$$\nabla\left(\frac{1}{n}\left\|X(I - PT_kP^T)\right\|_F^2\right) = -\frac{2}{n}(XP)^T(X - XPT_kP^T)P. \tag{12}$$

The detailed calculations are deferred to Section A of the Appendix.

Algorithm 1: Graph structure learning - minimization of (9)

Input: $\lambda, L, \epsilon > 0$.
Initialization: T_0 the null squared $p \times p$ matrix, $k = 0$ and $e = +\infty$.
while $e > \epsilon$ **do**
  Compute $U_k = T_k - \frac{\nabla\left(\frac{1}{n}\left\|X(I - PT_kP^T)\right\|_F^2\right)}{L}$ with Equation (12);
  Using Equation (10), compute the current matrix $T_{k+1} = \left((T_{k+1})_i^j\right)_{i,j}$;
  Project T_{k+1} on $\mathbb{T}_p(\mathbb{R})$ with Equation (11): $T_{k+1} \leftarrow \text{Proj}_{\mathbb{T}_p(\mathbb{R})}(T_{k+1})$;
  Compute $e = \|T_{k+1} - T_k\|_F$;
  Increase k: $k \leftarrow k + 1$;
end
Output: $T_k \in \mathbb{T}_p(\mathbb{R})$ the unique solution of (9).

4.3 A Genetic Algorithm for a global exploration of the permutation matrices space blending network topologies

As the optimal T can be calculated for any P using Algorithm 1, the optimization task (6) comes down to exploring the $\mathbb{P}_p(\mathbb{R})$ space of permutation matrices in dimension p. We first note that the number of permutation matrices is $p!$, which rules out any exact method, even with relatively small p. We propose instead to use a meta-heuristic approach, which has proven to be successful for many discrete optimization problems like wire-routing, transportation problems or traveling salesman problem (Michalewicz, 1994; Dréo, 2006).

Among the different meta-heuristics (Simulated annealing, Tabu search, Ant Colony,...) we have favored Genetic Algorithms (GA) because, despite limited convergence results (Cerf, 1998; Michalewicz, 1994), it has been found much more efficient in problems related to ours than alternatives with more established convergence proofs (e.g. Granville et al. (1994) for simulated annealing), while allowing the use of parallel computation.

GAs mimic the process of natural evolution, and use a vocabulary derived from natural genetics: populations (a set of potential solutions of the optimization problem), individuals (a particular solution) and genes (the components of a potential solution). In short, a population made of N potential solutions of the optimization problem samples the search space. This population is sequentially modified, with the aim of achieving a balance between exploiting the best solutions and exploring the search space, until some termination condition is met.

We use here a classical GA, as described in Michalewicz (1994) for instance, which is based on three main operators at each iteration: selection, crossover and mutation. The population is reduced by selection; selection shrinks the population diversity based on the individual fitness values. The crossover allows the mixing of good properties of the population to create new composite individuals. Mutations change one (or a few in more general GAs) components of the individuals to allow random space exploration. The complete sketch of the algorithm is given in Algorithm 2. The details of the different operators are given in the following.

Variables encoding As we show in Example 3, any $P \in \mathbb{P}_p(\mathbb{R})$ is uniquely defined by a permutation vector of $[\![1, p]\!]$. Hence, we use as a the search space $\mathfrak{S}_p$ the set of permutations of $[\![1, p]\!]$, which is a well-suited formulation for GAs.

Example 3. *Consider the permutation matrix ($p = 5$):* $P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$. *Then, P is represented by the* $\boxed{1}\boxed{5}\boxed{4}\boxed{3}\boxed{2}$ *vector, looking at the ranks of non-null values of P column by column. The nodes are ranked according to their topological ordering.*

Note that our problem closely resembles the classical traveling salesman problem (TSP), which has been succesfully addressed by means of genetic algorithms (Grefenstette et al., 1985; Davis et al., 1991). Identically to the TSP, we optimize over the space of permutations, which induces specific constraints for defining the crossover and mutation operators. However, unlike the TSP, the problem is not circular (in the TSP, the last city is connected to the first one), and the permutation here defines a hierarchy between nodes rather than a path, which makes the use of TSP-designed operators a potentially poor solution. As we show in the following, we carefully chose those operators in order to respect the nature of the problem at hand.

Fitness function Given a potential solution $p_i \in \mathfrak{S}_p$, the fitness function is defined as:

$$J_i = J(p_i) = \frac{1}{n} \left\| X(I - P_i T_i^* P_i^T) \right\|_F^2 + \lambda \left\| T_i^* \right\|_1 ,$$

with P_i constructed from p_i as in Example 3 and T_i^* the solution of Equation (9) with $P = P_i$. Hence, at each step of the proposed GA, the evaluation of the fitness function requires running the nested loop of our global algorithm.

Selection operator The selection operator (or survival step) consists in generating a population of N individuals from the N existing individuals by random sampling (with replacement, hence some individuals are duplicated and others are deleted). It aims at improving the average quality of the population by giving to the best potential solutions a higher probability to be copied in the intermediate population. We have chosen to use the classical proportional selection of Holland (1992): each individuals is selected with a probability proportional to its fitness value.

Crossover operator A crossover operator generates a new set of potential solutions (children) from existing solutions (parents). Crossover aims at achieving at the same time (i) a good exploration of the search space by mixing the characteristics of the parents to create potentially new ones while (ii) preserving some of the characteristics of the parents (good solution features). The crossover population (set of parents) is obtained by selecting each individual of the population with a probability p_{xo}; the parents are then paired randomly.

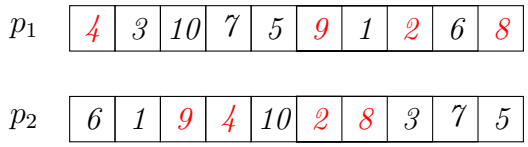
We have chosen the *order-based* crossover, originally proposed for the TSP (Michalewicz, 1994, Chapter 10), which is defined as follows. Given two parents p_1 and p_2, a random

set of crossover points are selected, which we denote Ω. It consists in a k-permutation of $[\![1, p]\!]$, with k uniformly drawn between 0 and p. A first child C_1 between p_1 and p_2 is then generated by:

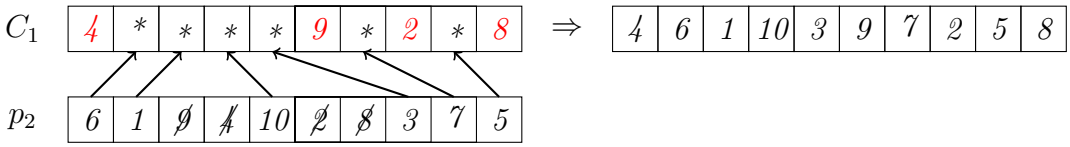
1. fixing the crossover points of p_1,
2. completing C_1 with the missing numbers in the order they appear in p_2.

A second child C_2, complementary of C_1, is created with the same procedure, replacing p_1 with p_2 (see Example 4).

Example 4. *Consider the two following parents:*



Assume that the crossover points randomly chosen are 4, 9, 2 *and* 8 *(in red above). Then, the child C_1 is defined by inheriting those points from p_1 and filling the other points in the order they appear in p_2:*



From a graphical point of view, a crossover between p_1 and p_2, which encode two complete graphs $\mathcal{G}_{P_1}$ and $\mathcal{G}_{P_2}$, constructs two new graphs. One of them, $\mathcal{G}_{C_1}$ is composed of the sub-graph of $\mathcal{G}_{P_1}$ induced by the set of crossover points Ω and the sub-graph of $\mathcal{G}_{P_1}$ induced by the complementary set Ω^C of Ω in $[\![1, p]\!]$ (see Figure 4). The second child graph $\mathcal{G}_{C_2}$ is obtained in an identical manner by reversing the roles played by the parent graphs.

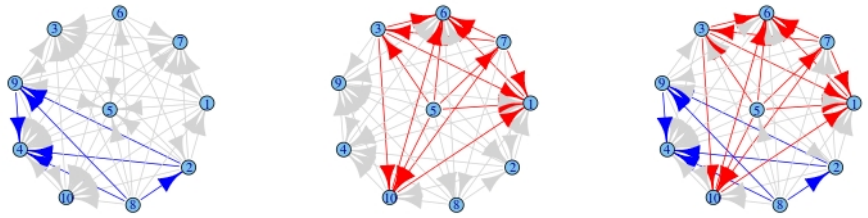
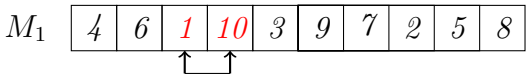


Figure 4: Graphical representation of crossover between two 10-node graphs (in red and blue above).

Mutation Mutation operators usually correspond to the smallest possible change in an individual (unary operator). We thus define it as an alteration of two neighbouring genes (see Example 5). Graphically, a mutation consists in switching the arrowhead of an edge between two nodes. Mutation is applied to each child with probability p_m.

Example 5. *A possible mutation for the first child of Example 4 is to swap the genes "1" and "10" (in red below):*



Stopping criterion Two quantities are monitored along the iterations: the heterogeneity of the population and the value of the objective function.

For the first indicator, we use the Shannon entropy, defined for each rank position $j \in [\![1, p]\!]$ as:

$$H_j = -\sum_{i=1}^{p} \frac{N_{i,j}}{N} \log\left(\frac{N_{i,j}}{N}\right),$$

where $N_{i,j}$ is the number of times when i appears in position j. $H_j = 0$ if all the individuals "agree" on the position of a node. On the contrary, it is maximum when we observe a uniform distribution of the different nodes at a given position. The algorithm stops if the population entropy value $H = \sum_{j=1}^{N} H_j$ drops below a threshold since $H = 0$ if all the individuals are identical. A second criterion can terminate the GA if difference in the average fitness (denoted $\bar{J}$ thereafter) of the population between four consecutive iterations, does not change by more than a predefined threshold.

Algorithm 2: Algorithm overview

Input: $p_{xo}, p_m, \epsilon_H > 0, \epsilon_J > 0$, λ, L.
Initialization: Generate the initial population $\mathcal{P}_0$ with N permutations of $[\![1, p]\!]$, $k = 0$ and $e_J = +\infty$.
while $H > \epsilon_H$ & $e_J > \epsilon_J$ **do**
  Generate $\mathcal{P}_{k+1}$ as a random **selection** of N individuals from $\mathcal{P}_k$;
  Pick an even subset $\mathcal{P}_{xo}$ of $\mathcal{P}_{k+1}$ (each individual of $\mathcal{P}_{k+1}$ selected with probability p_{xo});
  Perform **crossover** on $\mathcal{P}_{xo}$ by randomly pairing the individuals;
  Mutate each obtained individual with probability p_m ;
  Evaluate the new individuals $\mathcal{P}_m$ by running Algorithm 1;
  Replace $\mathcal{P}_{xo}$ by $\mathcal{P}_m$ in $\mathcal{P}_{k+1}$;
  Compute the Shannon entropy H and the difference in the average fitness $e_J = \max_{0 \leq i \leq 4} \left(\bar{J}(\mathcal{P}_{k+1}) - \bar{J}(\mathcal{P}_{k-i})\right)$;
  Increase k: $k \leftarrow k + 1$;
end

5 Numerical experiments

This section is dedicated to experimental studies to assess practical performances of our method through two kinds of datasets. In a first phase, the aim of these applications is to

show that the global algorithm we propose has a sound behavior on a simulated toy data. In a second phase, we demonstrate the ability of our algorithm to analyse data sets, which have features encountered in real situations, and we compare it to other state-of-the art methods, namely the Bootstrap Lasso (Bach, 2008) and the Random Forests (Huynh-Thu et al., 2010). The competing methods are presented in Section 5.3.1 and Section 5.3.2 introduces the measures we used to assess the merits of the different methods. In Section 5.1, we present the calibration of the Genetic Algorithm parameters. Experimental results are then detailed in Section 5.2 for the simulated toy dataset, while Section 5.3 consists in the study of datasets which mimic the activity of a complex biological system. These latter datasets were used in a Machine Learning challenge (DREAM4, Marbach et al. 2009b).

5.1 Algorithm parameters

Running the procedure of Algorithm 2 requires to define parameters of the outer loop (choice of P) and of the nested loop (optimal T^*). The evaluation of the Lipschitz gradient constant L, used to find the optimal graph structure T^*, is known as a hard well-studied problem in optimization. Some authors propose to choose an estimate of L from a set of possible values (Jones et al., 1993; Sergeyev & Kvasov, 2006), to estimate local Lipschitz constants (Sergeyev, 1995), or to give it a priori (Evtushenko et al., 2009; Horst & Pardalos, 1995). Here, observing Equation (12), a major bound for L is given by:

$$L \leq \frac{2}{n} \left\| X^T X \right\|_F .$$

We found that setting L to this bound worked well in practice in all our scenarii.

Five parameters need to be tuned to run the Genetic Algorithm: the crossover rate p_{xo}, the mutation rate p_m, the constant of the stopping criteria ϵ_H and ϵ_J and the size of the population N. For the first four parameters, we observed that their value had a limited effect on the efficiency, hence we chose commonly used values in the literature (see Table 1). The size of the population has a more complex effect and has been investigated in several prospective papers (e.g. Schaffer et al. 1989; Alander 1992; Piszcz & Soule 2006; Ridge 2007) but without providing a definitive answer to the problem. In our simulation study, we chose as a rule-of thumb $N = 5p$, which was found as a good compromise between computational cost and space exploration on several experiments.

The complete parameter settings used in our experiments are reported in Table 1.

Parameter	p_{xo}	p_m	N	L	max. nb. of eval.	ϵ_H	ϵ_J
Value	0.25	0.5	$5 \times p$	$\frac{2}{n} \left\| X^T X \right\|_F$	$5 \times p$	10^{-6}	10^{-4}

Table 1: Algorithm parameter settings

5.2 Algorithm illustration on a toy dataset

We consider here a 50 node DAG with a hierarchical shape: a first node is connected to five nodes, each of those being connected to nine nodes. The non-zero parameters $(G_0)_j^i$ of the matrix associated to the true DAG (shown in Figure 5) are uniformly sampled between 0.3 and one. Using this graph, we generate $n = 100$ observations following the hypotheses of

Section 2.1 (Gaussian, homoscedastic and centred error). We then run our global algorithm on this simulated data set.

For this graph, the true permutation with our parametrization is $p^* = (50, 49, \ldots, 1)$, but many permutations are also correct: node 1 must be at the last position, and nodes 2 to 6 only need to be placed after the nodes they dominate. To illustrate the behavior of our algorithm, we set the penalization parameter to 0.5 and we focus on the evolution of the following quantities:

- the value of the fitness function (as in Equation 6) of the current best solution and of the current population (Figure 6),
- the ranks of the nodes $1-6$ in the current permutation associated to the best solution (Figure 7), i.e. the position where they appear in the permutation,
- the Shannon entropy of each node in the current population (Figure 8), i.e. the diversity of the position where they appear in the permutations.

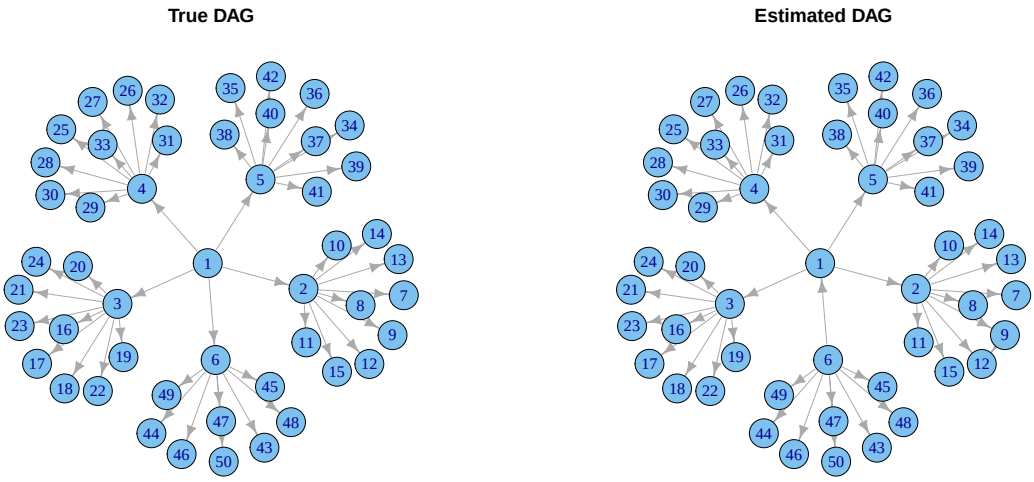


Figure 5: True DAG (left) and its estimation by our Algorithm 2 (right). Critical nodes are labelled 1 to 6: 1 is the root, 2 to 6 are the second level nodes while nodes 7 to 50 are leaf nodes. The estimated graph is almost identical to the actual one, except for the arrow between nodes 1 and 6 (reverse direction) and an extra arrow between nodes 9 and 12.

We first observe that the estimated graph (Figure 5, right) is almost identical to the true one (Figure 5, left). Most of the fitness function evolution is achieved within the first 50 iterations/generations. Moreover, for this particular problem, many permutations are equivalent from the likelihood perspective. This partly explains why the best solution marginally changes until the end of the run (Figure 7), and why the Shannon entropy (Figure 8) does not converge to zero, meaning that the population remains diverse. Moreover, the mutation operator introduces a small amount of randomness in the population at each generation. More specifically, we observe that the Shannon entropy of the root node 1 approaches zero, while the other critical nodes $2-6$ have a smaller entropy than the

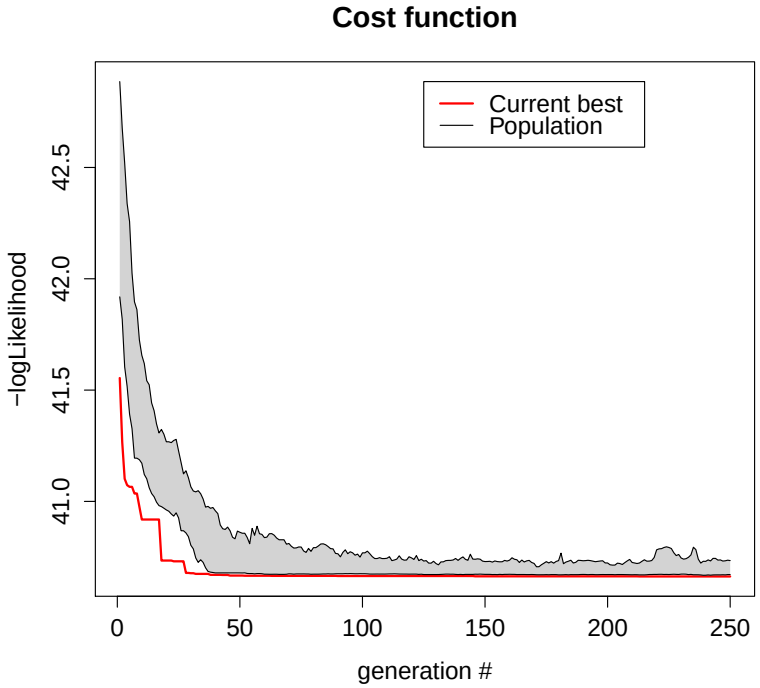


Figure 6: Evolution of the value of the fitness function of the current best solution (bold red line) and of the current population (the grey area shows the interval between the 10th and 90th percentile). Most of the cost function improvement is achieved after 30 generations. The population evolves towards small function values only but do not converge towards a unique value.

non-critical ones: they simply have fewer degrees of freedom in their position relative to non-critical nodes.

This first numerical experiment allows us to conclude that our algorithm show desirable features as expected on this toy problem: rapid convergence towards a set of good solutions with respect to the fitness while ensuring some diversity to guarantee a good exploration of the space of permutations. Moreover, the diagnostic quantity are good indicator to monitor the convergence of the proposed algorithm to find solutions to the minimization problem 6.

5.3 DREAM data analysis

The second type of datasets we used mimics activations and regulations that occur in gene regulatory networks. It is provided by the DREAM4 challenge on "In Silico Network Challenge" (Marbach et al., 2009a). Note that although plausibly simulated, DREAM4 data sets are not real biological data sets. However, the used network structures (5 in total) were extracted from *E. coli* and *S. cerevisae* -2 biological model organisms- trancriptional networks. Note that these networks contain cycles, but self-loops were discarded. The gene expression observations were not simulated by an equal noise Gaussian multivariate model. On the contrary, stochastic differential equations were used to mimic the kinetic laws of intricate and intertwined gene regulations. In addition to the biological noise simulated from the stochastic differential equations, technical noise was added to reproduce actual

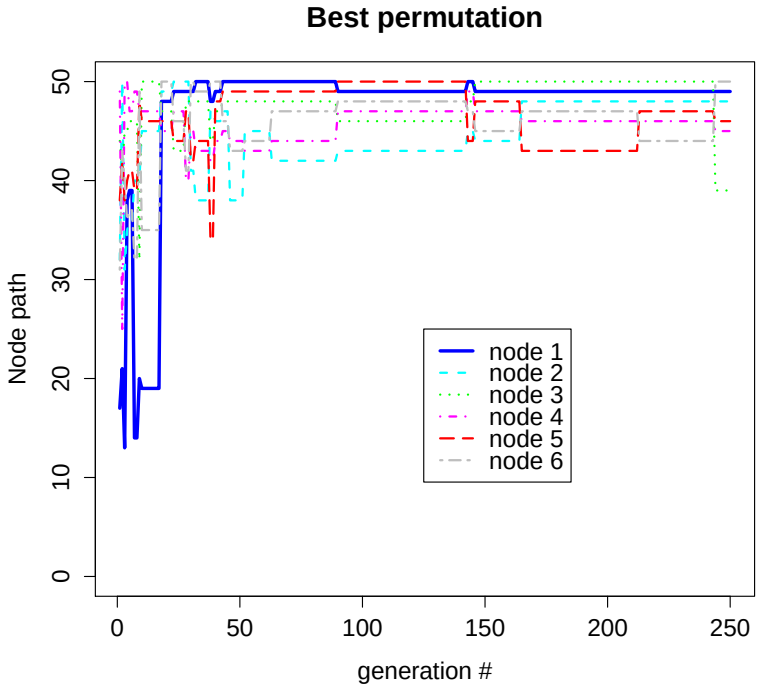


Figure 7: Evolution of the ranks of the critical nodes 1 to 6 of the current best solution. All the nodes evolve towards high ranks within the first 30 iterations. The node 1 takes the rank 49 instead of 50, which explains the inversion of the arrow between node 1 and 6 (as pointed out Figure 5 caption).

gene measurement noise. All data sets were generated by the `GNW` software (Marbach et al., 2009b).

Working with simulated networks, we are able to quantitatively and objectively assess the merit of different methods in terms of true positive vs. false positive (noisy predictions) and false negative (incomplete predictions) edges. While the analysis of a real data set is certainly the final goal of a methodology motivated by a real problem like ours, there are only imprecise ways of validating a method when analysing a real data set. Well known systems are often small and even if knowledge has accumulated on them, these can be noisy and difficult to gather to obtain a fair picture of what can adequately be considered as sets of true positive and true negative sets of edges. Even if the data generation process of the DREAM4 In Silico Network Challenge is completely understood, no existing method is able to predict all regulatory relationships, but at the price of including many false positive predictions. In addition, the DREAM4 datasets we considered have $p = 100$ nodes and only $n = 100$ observations making it a a very challenging task.

5.3.1 Comparison to state-of-the art

We compare our Genetic Algorithm to other state-of-the art inference methods. These methods decompose the prediction of the network into p feature selection sub-problems. In each of the p sub-problems, one of the node is predicted from the other ones using random forests (Breiman, 2001) or a bootstrapped version of the Lasso (Tibshirani, 1996), denoted BootLasso thereafter. Random forests obtained the best performing rank on the DREAM4

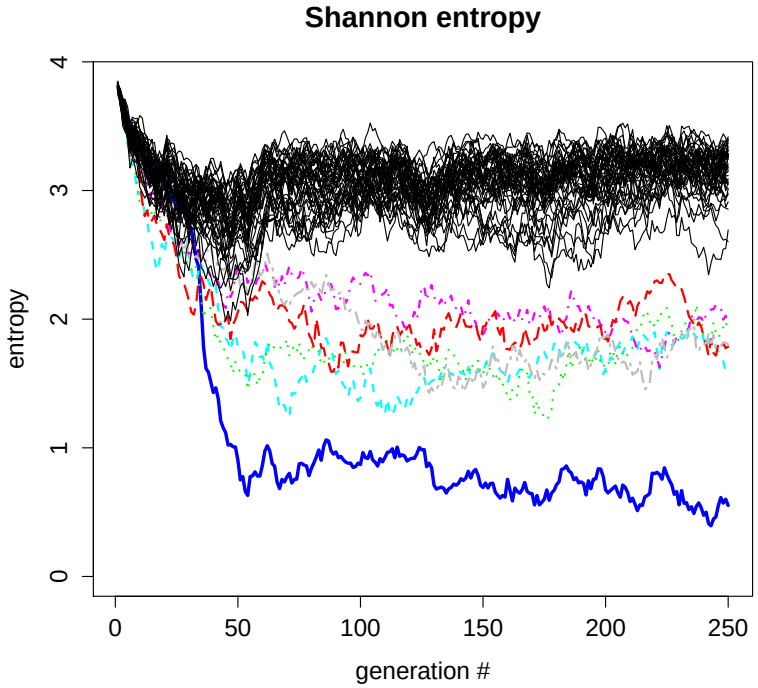


Figure 8: Evolution of the Shannon entropy of the critical nodes 1 to 6 and of the others within the current population. The legend is the same as in Figure 7; black curves stand for nodes 7 to 50.

In Silico Networks challenge Huynh-Thu et al. (2010) while the Bootstrap Lasso allowed a meta-analysis to achieve even better performances than the best performing team of the DREAM5 Systems Genetics challenge (de la Fuente & Stolovitzky, 2010; Vignes et al., 2011; de la Fuente, 2013).

The Lasso is a ℓ_1-norm penalization technique for solving linear regression. Following the works of Bach (2008), BootLasso uses bootstrapped estimates of the active regression set based on a Lasso penalty: only those variables that are selected in every bootstrap are kept in the model, and actual coefficient values are estimated from a straightforward least square procedure. Note that we slightly relax the condition for a variable to be included in the model. A variable was selected at a given penalty level if more than 80% of bootstrapped samples led to selecting it in the model. This strategy was observed to be more efficient in a different but related setting (de la Fuente, 2013). We used 100 bootstraps in our numerical experiments and used all penalty values that detected a change in the regression set for each of the sub-problems. As a last step, we refitted the finally predicted model with least square estimates to obtained unbiased estimates of the regression coefficients. Our implementation was based on the R package `glmnet` (Friedman et al., 2010).

For the random forest approach, each gene expression was successively considered as a target, and the method sought regulators of that gene in the expressions of all other genes. More specifically, regulators were detected as most significant explanatory variables according to a variance reduction criterion in a regression tree framework. The process was repeated on a randomized ensemble of trees, which made up the so-called random forest. This method allowed us to derive a ranking of the importance of all regulator expressions for the target by averaging the scores over all the trees of the random forest. We only kept

those scores which were higher than the cut-off score of a random pseudo-gene included in each regression tree. The randomized subset of regulators tested at each tree split allowed us to avoid local minima of the global score. Finally, the random sub-sample of the data, we used for each tree avoided over-fitting of the data. Robust estimates were obtained by repeating the analysis of each forest 100 times with 100 different random pseudo-gene. Our implementation was based on the R package `randomforest` (Liaw & Wiener, 2002) with $1,000$ tress grown in each forest. Other parameters were kept to their default value, in particular the number of nodes tested at each split was $\lfloor p/3 \rfloor$.

5.3.2 Performance metrics

A classical performance measure for graph inference methods consists in comparing predicted interactions with the known edges in the true graph $\mathcal{G}_0$ using precision versus recall (P/R) curves. We denote TP, respectively FP, FN and TN, the true positive (correctly predicted) edges, respectively the false positive (inferred by mistake) edges, the false negative (missed) edges, and the true negative (correctly non-predicted) edges. The recall, defined as $\frac{TP}{TP+FN}$, measures the power (or sensitivity) of reconstruction of non-zero elements of the true matrix G (or equivalently of the true network) for one method, whereas the precision, equal to $\frac{TP}{TP+FP}$, measures the accuracy of the reconstruction. The closer to 1 the precision and the recall the better.

P/R curves represent the evolution of those quantities when varying the sparsity of the methods. Random forests produce as an output a ranked list of regulatory interactions, which corresponds to the edges of the inferred graph. Edges are then successively introduced with decreasing confidence scores to produce the random forest P/R curve. Contrary to the random forest algorithm, our proposed Genetic Algorithm and BootLasso are based on penalized optimization: both seek linear dependencies between the variables with a controlled level of parsimony (λ in Equation (5) for the Genetic Algorithm). For λ varying from 0 (complete graph) to $+\infty$ (empty graph), each of these methods produce a list of edges, successively introduced in the model. These lists of edges define the precision versus recall curves for these two approaches.

As a summary performance measurement, we also computed the classical area under the P/R curve (AUPR).

5.4 Numerical results

The P/R curves for the five DREAM problems are shown in Figure 9. Each curve corresponds to one of the five networks used in the challenge. In general, for all the problems the three methods are able to achieve a precision equal to 1 (that is, to include only true edges), but these correspond to overly sparse graphs (very small recall). Conversely, a recall equal to 1 can only be reached by adding a large number of FP edges. The main differences between the methods appear on the leftmost part of the P/R curves, especially B, C and D: while the precision of BootLasso and random forests drops rapidly with a slow increases in recall above 20% recall, it remains higher for the GA. Hence, its first predicted edges are at least as accurate than those of the two other methods and it produces a larger set of reliable edges. For graphs of lesser sparsity, none of the three methods is really able to identify clearly reliable edges. Large number of FP edges are produced to achieve a recall higher than 60%.

For Networks 1 and 5 (Figure 9 A and E), the GA recovers with more difficulty the first true edges than BootLasso and Random Forests, with a high level of FP edges at the beginning of the curve (low precision and low recall). However, as soon as the recall exceeds the 10%, *resp.* 15%, for graph A, *resp.* for graph E, the GA is again better than other methods.

In addition, Table 2 gives the areas under the P/R curves for all methods and networks. For this indicator, GA significantly outperforms the state-of-the-art methods for all networks.

Method	Genetic Algorithm	BootLasso	Random Forests
Network 1	0.182	0.118	0.154
Network 2	0.236	0.061	0.155
Network 3	0.348	0.171	0.231
Network 4	0.317	0.147	0.208
Network 5	0.267	0.169	0.197

Table 2: Area under the Precision vs. Recall curve for all networks and methods.

6 Conclusion and remarks

In this paper, we proposed a hybrid genetic/convex algorithm for inferring large graphs based on a particular decomposition of the ℓ_1-penalized maximum likelihood criterion. We obtained two oracle inequalities that ensure that the graph estimator converges to the true graph under assumptions that mainly control the model structure: graph size (balance between sparsity, number of nodes and maximal degree) and signal-to-noise ratio. From an algorithmic point of view, the estimation task decomposes into two subproblems: the node ordering estimation and the graph structure learning. The first one is a non-trivial problem since we optimize over a discrete non-convex large dimensional set, which led us to use a heuristic approach. The second one is a more common problem, related to the Lasso one, for which we proposed a tailored procedure. The potential of such an approach clearly appeared in the numerical experiments, for which the behavior of our algorithm seemed to be competitive compared to the state-of-the art.

Nevertheless, we see many opportunities for further improvements. First, convergence proof for the algorithm, although a challenging task, is worth investigating, for instance using the works of Cerf (1998) on genetic algorithms. An alternative would be to consider other optimization schemes for the node ordering with more established convergence proofs (e.g., Granville et al., 1994, for simulated annealing).

Second, other potential extensions involve algorithmic considerations in order to improve the calculation time, including a finer calibration of the algorithm parameters, an initialization step for the gradient descent, and in general increasing the communication between the nested and outer loops. Tackling very large datasets (from several thousands of nodes) may also require a particular treatment, for instance by adding local search operators to the genetic algorithm.

Finally, we would like to emphasize the graph identifiability problem: in our settings, we assume the noise variances of all graph nodes to be equal to ensure graph identifiability

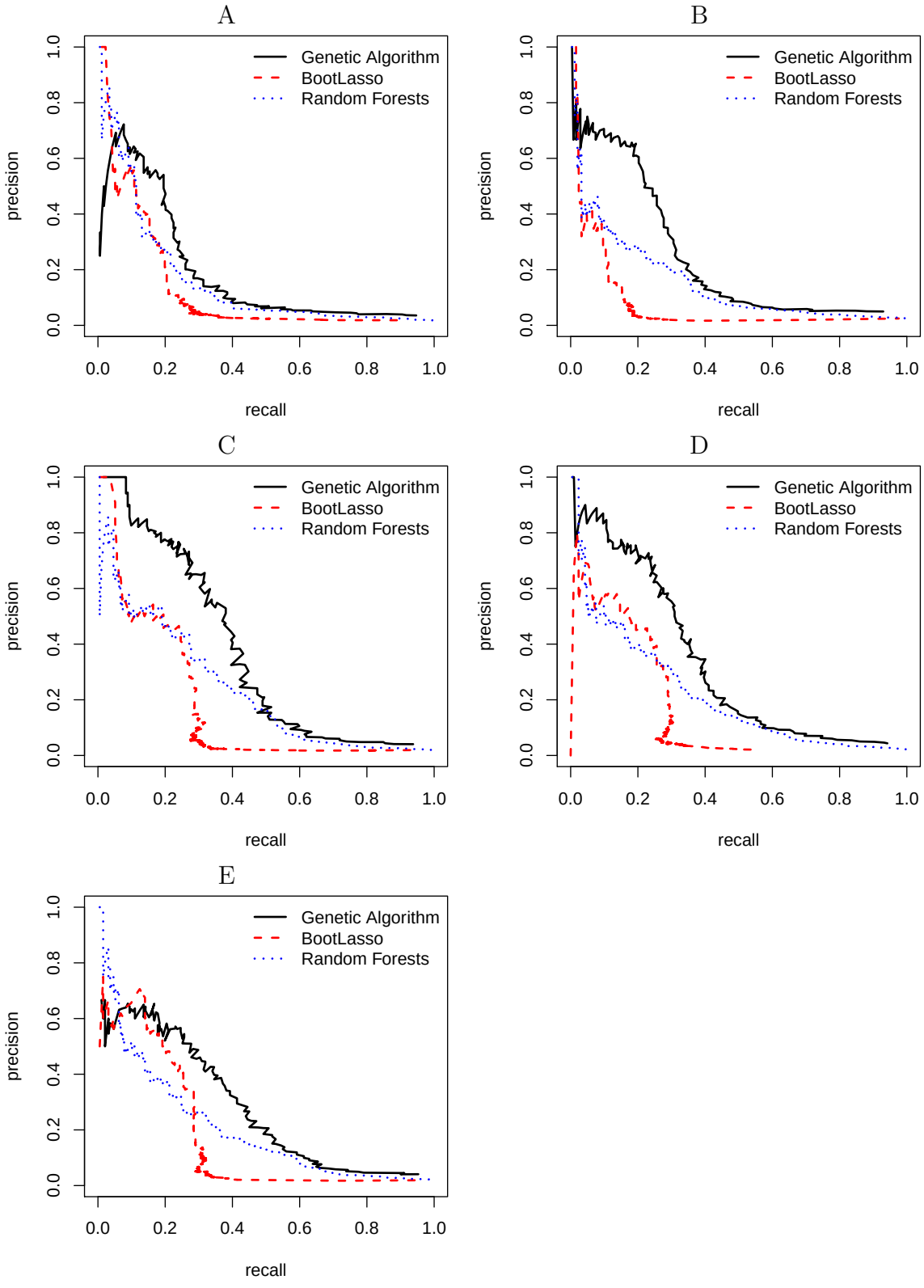


Figure 9: P/R curves for the five Dream networks and the three compared methods.

(that is no equivalence class of graphs). Such a hypothesis is of course restrictive and likely to be violated for real datasets. In order to infer networks for any noise variances, one solution consists in incorporating interventional data on the model. These data are

obtained from perturbations of the system (like gene knockouts) and make the equivalence class of graphs smaller (Hauser & Bühlmann, 2012). Then, the use of such new data could be combined with observational on the MLE estimator (as recently proposed by Hauser & Bühlmann (2015) for a BIC-score penalized MLE, or by Rau et al. (2013) for learning Gaussian Bayesian networks in the case of GRN inference) and a modification of our hybrid algorithm could lead to the identification of the true graph.

A Calculation details of Section 4.2

In this section, we present the detailed calculations of Section 4.2. These should ensure that a solution of Equation (9) is given by performing Algorithm 1. The objective function in Equation (9) can be split into a sum of two functions: a convex, L-smooth part $g(T) = \frac{1}{n}\left\|X - XPTP^T\right\|_F^2$ and a penalization term $h(T) = \lambda \|T\|_1$. As a L-smooth function, g is differentiable, its gradient is L-Lipschitz-continuous and a standard convex analysis result (Hiriart-Urruty & Lemaréchal, 1993) provides:

$$g(T) \leq g(U) + \langle \nabla g(U), T - U \rangle_F + \frac{L}{2} \|T - U\|_F^2 ,$$

for all $p \times p$ matrices T and U, where $\langle ., . \rangle_F$ stands for the inner product associated to the Frobenius norm.

A natural idea to minimize the $g + h$ function consists in defining a sequence $(T_k)_{k\geq 0}$ such that:

$$T_{k+1} = \underset{T}{\operatorname{argmin}} \left\{ g(T_k) + \langle \nabla g(T_k), T - T_k \rangle_F + \frac{L}{2} \|T - T_k\|_F^2 + h(T) \right\}, \tag{13}$$

which ensures that the sequence $(g(T_k) + h(T_k))_k$ decreases. Then, adding $\frac{L}{2}\left\|\frac{\nabla g(T_k)}{L}\right\|_F^2$ (which does not depend on T) to the minimization problem (13), a standard equality between sums of squared norms yields:

$$T_{k+1} = \underset{T}{\operatorname{argmin}} \left\{ \frac{L}{2} \left\| T - \left(T_k - \frac{\nabla g(T_k)}{L} \right) \right\|_F^2 + h(T) \right\}. \tag{14}$$

Denote $U_k = T_k - \frac{\nabla g(T_k)}{L}$. Equation (14) then becomes:

$$T_{k+1} = \underset{T}{\operatorname{argmin}} \left\{ \frac{L}{2} \|T - U_k\|_F^2 + \lambda \|T\|_1 \right\}.$$

In an element-wise formulation, this writes:

$$(T_{k+1})_i^j = \underset{T_i^j \in \mathbb{R}}{\operatorname{argmin}} \left\{ \frac{L}{2} \left(T_i^j - (U_k)_i^j \right)^2 + \lambda \left| T_i^j \right| \right\}, \tag{15}$$

for all $i, j \in [\![1, p]\!]$.

Lemma 1 below allows us to derive an explicit solution to Equation (15). It follows from technical calculations we don't detail here.

Lemma 1. *Denote $\varphi(x) = (x - x_0)^2 + \frac{2\lambda}{L}|x|$, with $x_0 \in \mathbb{R}$ and $\lambda, L \in (\mathbb{R}^+)^2$. A solution of the optimization problem $\min_{x\in\mathbb{R}} \varphi(x)$ is given by:*

$$x = \operatorname{sign}(x_0) \max\left(0, |x_0| - \frac{\lambda}{L}\right),$$

where sign(.) *is the notation for the sign of any non-zero element.*

From Lemma 1, we deduce that a solution of (15) is:

$$(T_{k+1})_i^j = \operatorname{sign}\left((U_k)_i^j\right) \max\left(0, \left|(U_k)_i^j\right| - \frac{\lambda}{L}\right),$$

where $U_k = T_k - \frac{\nabla g(T_k)}{L}$. This concludes the justification for Algorithm 1.

B Proof of Theorem 1

Outline of the proof: in section B.1, we introduce the notations and present some technical lemmas. We also propose a discussion around Assumption $\mathbf{H_3}$. The proof of Theorem 1 then works as follows: in section B.2, we investigate what happens when estimating a wrong permutation $\hat{P} \notin \Pi_0$. We present a bound for $\sum_j \left(|\hat{\omega}_j|^2 - 1\right)^2$, with $\hat{\omega}_j$ the variance of $\varepsilon^j(\hat{P})$, on a probability space where the random components behave well. We show that this event has a large probability to happen and we obtain a contradiction of Assumption $\mathbf{H_{id}}$. Then, given a true order of variables $\hat{P} \in \Pi_0$, in Section B.3, we present two bounds for the estimator $\hat{G}$ by careful manipulations.

B.1 Notations and technical results

We begin by recalling the model structure and some induced technical results. Let us first introduce useful graph terminology: in a graph $\mathcal{G}$ made of p vertices $V = \{X^1, ..., X^p\}$ and a set of edges $E \subset V \times V$, X^i is said to be a parent of X^j if there is a directed edge between X^i and X^j ($X^i \to X^j$). More generally, X^i is a descendant of X^j if there exists a directed path (sequence of directed edges) between X^i and X^j. We denote by $\mathrm{Pa}(X^j)$, *resp.* $\mathrm{Des}(X^j)$, the set of parents, *resp.* descendants of X^j. Let also denote by $\mathrm{ND}(X^j)$, the complementary set of $\mathrm{Des}(X^j)$ (the non-descendants of X^j). We recall here the model we cast:

$$X = XG_0 + \varepsilon,$$

where $G_0 = \left((G_0)_i^j\right)_{1 \leq i,j \leq p}$ with G_0^j representing the p-vector of linear effects of all nodes on node X^j. More precisely, a non-zero entry $(G_0)_i^j$ of G_0 encodes a directed edge between X^i and X^j in the graph $\mathcal{G}_0$.

In this framework, lemma 2 below highlights independence between any non-descendant of X^j and the noise associated to X^j, for a given node X^j.

Lemma 2 (Peters et al. (2011))**.** $\forall X^i \subset \mathrm{ND}(X^j), \quad \varepsilon^j \perp X^i$.

A proof of this result is given in (Peters et al., 2011). As a consequence, we deduce that:

$$\varepsilon^j \perp X^i, \quad \text{as soon as } (G_0)_i^j \neq 0. \tag{16}$$

Remind that G_0 can be written as a strictly lower-triangular matrix after permutation of its rows and columns: $G_0 = P_0 T_0 P_0^T$, with $P_0 \in \Pi_0$ (the set of permutations compatible with the true DAG). For any permutation P, we denote by $G_0(P) := P P_0^T G_0 P_0 P^T$ the permuted graph (see Section 3.1 for more explanations), $\varepsilon(P) := X - XG_0(P)$ the $p \times p$ residual term associated to $G_0(P)$ and $\Omega(P)$ its covariance matrix. Lemma 3 below is then satisfied:

Lemma 3. *The variables $(\varepsilon^j(P))_{1 \leq j \leq p}$ are independent and the covariance matrix $\Omega(P)$ of $\varepsilon(P)$ is diagonal.*

Proof. A consequence for Lemma 2 is that, for all $j \in [\![1, p]\!]$, $\varepsilon^j(P) \perp \left(X^k\right)_{X^k \in \mathrm{Pa}(X^j)}$. Moreover, for any $X^k \in \mathrm{Pa}(X^j)$, X^k (and by extension $\varepsilon^k(P)$) can be written as a linear combination of $\left(X^{k'}\right)_{X^{k'} \in \mathrm{Pa}(X^k)}$. We thus deduce that $\varepsilon^j(P)$ is independent of everything used before. This implies that all error terms are independent and the covariance matrix

$\Omega(P)$ of $\varepsilon(P)$ is diagonal. Until the end of the proof, we denote by $\omega_j(P)$ its diagonal elements. □

Next lemma then provides a technical relation between the error ε^j associated to variable X^j and the permuted error $\varepsilon^j(P)$.

Lemma 4. $\|\varepsilon\|_F^2 = \sum_{j=1}^p \frac{\sum_{k=1}^n \left(\varepsilon_k^j(P)\right)^2}{\omega_j^2(P)}$.

Proof. The covariance matrix Σ of X is given by:

$$\Sigma = [(I - G_0)^{-1}]^T (I - G_0)^{-1} = [(I - G_0(P))^{-1}]^T \Omega(P) (I - G_0(P))^{-1}.$$

We thus have:

$$\begin{aligned} \|\varepsilon\|_F^2 = \|X(I - G_0)\|_F^2 &= \text{trace}\left(X(I - G_0)\left(X(I - G_0)\right)^T\right) \\ &= \text{trace}\left(X(I - G_0(P))\Omega(P)^{-1}(I - G_0(P))^T X^T\right). \end{aligned}$$

Then, with $\Omega(P) = \text{diag}(\omega_1^2(P), ..., \omega_p^2(P))$ (see Lemma 3), we deduce:

$$\|\varepsilon\|_F^2 = \sum_{i,j} \left((X(I - G_0(P)))_i^j\right)^2 \frac{1}{\omega_j^2(P)} = \sum_{j=1}^p \frac{\sum_{k=1}^n \left(\varepsilon_k^j(P)\right)^2}{\omega_j^2(P)},$$

which ends the proof. □

Lemma 5 states that under Assumption $\mathbf{H_{2-3}}$, the minimal eigenvalue of the covariance matrix Σ of X is controled while p (and n) grows to infinity.

Lemma 5. *Denote by λ_{min} the minimal eigenvalue of the covariance matrix Σ of X. Then, the following bound holds for λ_{min}:*

$$\lambda_{min} \geq \frac{1}{p \max\left(1, \|G_0\|_\infty^2\right)(1 + s_{max})},$$

with s_{max} as in $\mathbf{H_3}$.

Proof. Since P_0 is orthogonal and T_0 is strictly lower triangular, $\det(I - G_0) = \det(P_0(I - T_0)P_0^T) = 1$, and $\det(\Sigma) = [(I - G_0)^{-1}]^T (I - G_0)^{-1} = 1$.

Let $\chi_\Sigma(\lambda)$ be the characteristic polynomial of Σ and denote by $(\lambda_1, ..., \lambda_p)$ its p non-negative eigenvalues. On the one hand, $\chi_\Sigma(\lambda) = \prod_{i=1}^p (\lambda - \lambda_i)$ and

$$\chi'_\Sigma(0) = (-1)^{p-1} \sum_{i=1}^p \prod_{j \neq i} \lambda_j.$$

A minor bound for $|\chi'_\Sigma(0)|$ is $\left|\prod_{j \neq i} \lambda_j\right|$ for a given $i \in [\![1, p]\!]$. In particular, considering the index i that corresponds to the smallest eigenvalue and using $\det(\Sigma) = 1$, we obtain:

$$|\chi'_\Sigma(0)| \geq \left| \prod_{\lambda_j \neq \lambda_{min}} \lambda_j \right| = \frac{1}{\lambda_{min}}. \tag{17}$$

On the other hand, for a given matrix $M \in \mathcal{M}_{p\times p}(\mathbb{R})$, the derivative χ'_M of the characteristic polynomial of M is given by (Petersen & Pedersen, 2012):

$$\chi'_M(\lambda) = -\det(M-\lambda I)\,\mathrm{trace}\left((M-\lambda I)^{-1}\right).$$

Using $\det(\Sigma) = 1$ again, we thus have:

$$|\chi'_\Sigma(0)| = \mathrm{trace}(\Sigma^{-1}) = \|I - G_0\|_F^2 .$$

To finish the proof, note that the diagonal of G_0 is null:

$$\begin{aligned}\chi_\Sigma(0)' = \|I-G_0\|_F^2 &= \sum_{i=1}^{p} 1 + \sum_{i=1}^{p}\sum_{\substack{j=1\\ j<i}}^{p} \left((G_0)_i^j\right)^2 \\ &\leq p\left(1 + s_{max}\,\|G_0\|_\infty^2\right) \\ &\leq p\max\left(1, \|G_0\|_\infty^2\right)(1+s_{max}).\end{aligned}$$

This ends the proof with Equation (17). □

B.2 Estimation of the order of variables

We now turn to the proof of the first part of Theorem 1. To prove it, we assume that the permutation we estimate is a wrong permutation: $\hat{P} \notin \Pi_0$. For clarity purpose, until the end of the proof, we denote by $\hat{G}_0 := G_0(\hat{P})$, $\hat{\varepsilon} := \varepsilon(\hat{P})$ and $\hat{\omega} := \omega(\hat{P})$. A standard norm equality yields:

$$\left\|X\hat{G} - X\hat{G}_0\right\|_F^2 = \left\|X - X\hat{G}\right\|_F^2 - \left\|X - X\hat{G}_0\right\|_F^2 + 2\langle X - X\hat{G}_0, X\hat{G} - X\hat{G}_0\rangle_F,$$

where $\hat{G}$ is defined as (see Equation (5)),

$$\frac{1}{n}\left\|X - X\hat{G}\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 \leq \frac{1}{n}\|X - XG_0\|_F^2 + \lambda\|G_0\|_1 .$$

Then, Lemma 4 implies:

$$\begin{aligned}\frac{1}{n}\left\|X\hat{G} - X\hat{G}_0\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 &\leq \frac{1}{n}\sum_{j=1}^{p}\frac{\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2} + \lambda\|G_0\|_1 - \frac{1}{n}\|\hat{\varepsilon}\|_F^2 + \frac{2}{n}\langle\hat{\varepsilon}, X\hat{G} - X\hat{G}_0\rangle_F \\ &\leq \frac{1}{n}\sum_{j=1}^{p}\left(\frac{1}{\left|\hat{\omega}_j\right|^2} - 1\right)\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2 + \frac{2}{n}\langle\hat{\varepsilon}, X\hat{G} - X\hat{G}_0\rangle_F + \lambda\|G_0\|_1 ,\end{aligned}$$

and we finally obtain:

$$\begin{aligned}
\frac{1}{n}\left\|X\hat{G}-X\hat{G}_0\right\|_F^2+\lambda\left\|\hat{G}\right\|_1 &\leq \sum_{j=1}^{p}\frac{|\hat{\omega}_j|^2-\frac{1}{n}\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{|\hat{\omega}_j|^2}\left(|\hat{\omega}_j|^2-1\right)+\sum_{j=1}^{p}\left(1-|\hat{\omega}_j|^2\right)\\
&\quad+\frac{2}{n}\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F+\lambda\left\|G_0\right\|_1\\
&\leq \underbrace{\sqrt{\sum_{j=1}^{p}\left(\frac{|\hat{\omega}_j|^2-\frac{1}{n}\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{|\hat{\omega}_j|^2}\right)^2}}_{=I}\sqrt{\sum_{j=1}^{p}\left(|\hat{\omega}_j|^2-1\right)^2}\\
&\quad+\underbrace{\sum_{j=1}^{p}\left(1-|\hat{\omega}_j|^2\right)}_{=II}+\frac{2}{n}\underbrace{\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F}_{=III}+\lambda\left\|G_0\right\|_1. \quad (18)
\end{aligned}$$

Lemmas 6 and 7 below aim at bounding the terms I and III, with large probability.

Lemma 6. *Assume that Assumption* $\mathbf{H_{dim}}$ *is satisfied. Then, with probability at least* $1-\frac{2}{p}$*, there exists* $C_1>0$ *such that:*

$$\sum_{j=1}^{p}\left(\frac{|\hat{\omega}_j|^2-\frac{1}{n}\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{|\hat{\omega}_j|^2}\right)^2\leq C_1\frac{\log p}{n}\left(p+\hat{s}_0\right),$$

where $\hat{s}_0:=\sum_{i,j}\mathbb{1}_{\left(\hat{G}_0\right)_i^j\neq 0}$ *is the number of non-zero coefficients of* $\hat{G}_0$*.*

Proof of Lemma 6. The proof of this result is already given in van de Geer & Bühlmann (2013). For a better understanding, we recall key elements of the proof here. Denote by:

$$Z_j(P):=\frac{\frac{1}{n}\sum_{k=1}^{n}\left(\varepsilon_k^j(P)\right)^2-|\omega_j(P)|^2}{|\omega_j(P)|^2},$$

and assume that $G_0(P)$ is $s_0(P)$-sparse, *i.e.* $G_0(P)$ has $s_0(P)$ non-zero entries. Using Bernstein-like concentration inequalities, we can show that:

$$\mathbb{P}\left(\exists P,\sum_{j=1}^{p}Z_j(P)^2\geq 8\left(\frac{pt+(1+8\alpha)s_0(P)\log(p)+2p\log p}{n}\right)+8\left(\frac{4p(t^2+\log^2 p)}{n^2}\right)\right)\leq 2e^{-t},$$

for all $t\geq 0$, where α is some constant such that $p^4\leq\alpha n$. The conclusion then holds with $P=\hat{P}$ and $t=\log p$. □

Lemma 7. *Assume that Assumptions* $\mathbf{H_1}$ *and* $\mathbf{H_{dim}}$ *are satisfied. Then, with probability at least* $1-\frac{1}{p}$*, there exists* $C_3>0$ *such that:*

$$\frac{2}{n}\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F\leq C_3\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1,$$

where $\hat{s}_{0,j}$ *is the number of non-zero coefficients of the* p*-vector* $\hat{G}_0^j$*.*

Proof of Lemma 7. Remark that:

$$\langle \hat{\varepsilon}, X\hat{G} - X\hat{G}_0 \rangle_F = \sum_{i,j} \left(\hat{G} - \hat{G}_0\right)_i^j \sum_k X_k^i \hat{\varepsilon}_k^j.$$

To prove Lemma 7, we aim at showing that, uniformly over the set of permutation matrices and uniformly on $1 \leq i \leq p$, $\sum_k X_k^i \varepsilon_k^j(P)$ is bounded.

Let $(V_k)_{k=1,..,n}$ *i.i.d* random variables generated according to a $\mathcal{N}(0,1)$ distribution. A standard concentration inequality yields:

$$\mathbb{P}\left(\frac{1}{n}\sum_{k=1}^n V_k \geq t\right) \leq \exp(-nt^2).$$

Let $P \in \mathbb{P}_p(\mathbb{R})$. Denote by $\mathcal{A}_j(P)$ and $\mathcal{B}_j(P)$ the sets defined as:

$$\mathcal{A}_j(P) = \{\beta \in \mathbb{R}^p, \forall \beta_i \neq 0, X^i \perp \varepsilon^j(P), \quad \text{with } \varepsilon^j(P) = X^j - \textstyle\sum_k X^k \beta_k\}$$

$$\mathcal{B}_j(P) = \left\{ \exists \beta \in \mathcal{A}_j(P), \quad \frac{2}{n}\sum_k X_k^i \varepsilon_k^j(P) \geq 2\sigma_0 \frac{\sqrt{t + s_j(P)\log p + 2\log p}}{n} \right\},$$

where $\varepsilon^j(P) = X^j - \sum_k X^k \beta_k$ and $s_j(P)$ is the number of non-zero entries of some $\beta \in \mathcal{B}_j(P)$. Remark that for all j, $G_0^j(P) \in \mathcal{A}_j(P)$.

Under Assumption $\mathbf{H_1}$, the random variables $\varepsilon^j(P)$ follow a $\mathcal{N}(0, \omega_j^2(P))$, with $\left|\omega_j^2(P)\right| \leq \sigma_0^2$. We thus deduce that:

$$\mathbb{P}(\mathcal{B}_j(P)) \leq \exp\left(-\left(t + s_j(P)\log(p) + 2\log p\right)\right).$$

Let $m \in [\![1, p]\!]$. We now let P vary over all permutations such that $s_j(P) = m$, and we denote Π_m this set. On Π_m, node j has exactly m parents, and there exists at most $\binom{p}{m}$ possibilities for P. We then have:

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{P\in\Pi_m} \mathcal{B}_j(P)\right) &\leq \binom{p}{m} \exp\left(-\left(t + s_j(P)\log(p) + 2\log p\right)\right) \\
&\leq \exp\left(-\left(t + 2\log p\right)\right).
\end{aligned}$$

For m and j varying as possible, we conclude that:

$$\mathbb{P}\left(\bigcup_{j\in[\![1,p]\!]} \bigcup_{P\in\Pi} \mathcal{B}_j(P)\right) \leq \exp\left(-t\right).$$

Then, with probability at least $1 - e^{-t}$,

$$\begin{aligned}
\langle \hat{\varepsilon}, X\hat{G} - X\hat{G}_0 \rangle_F &\leq \sum_{i,j} 2\sigma_0 \sqrt{\frac{t + \hat{s}_{0,j}\log p + 2\log p}{n}} \left(\hat{G} - \hat{G}_0\right)_i^j \\
&\leq 2\sigma_0 \max_j \sqrt{\frac{t + \hat{s}_{0,j}\log p + 2\log p}{n}} \left\|\hat{G} - \hat{G}_0\right\|_1,
\end{aligned}$$

which ends the proof with $t = \log p$. □

The last term II of Equation (18) is bounded using inequality $\log(1+x) \leq x - \frac{1}{2(1+c_0^2)}x^2$, satisfied for $-1 \leq x \leq c_0$, to $x = |\hat{\omega}_j|^2 - 1$, which satisfies $-1 \leq x \leq \sigma_0^2 - 1$ (Hypotheses $\mathbf{H_1}$):

$$\sum_{j=1}^{p} \log\left(|\hat{\omega}_j|^2\right) \leq \sum_{j=1}^{p} \left(|\hat{\omega}_j|^2 - 1\right) - \frac{1}{2\sigma_0^4} \sum_{j=1}^{p} \left(|\hat{\omega}_j|^2 - 1\right)^2 .$$

Moreover, $\det(\Sigma) = \prod \hat{\omega}_j^2 = 1$ (see the proof of Lemma 5 for more details). We thus deduce that $\sum_j \log\left(|\hat{\omega}_j|^2\right) = 0$ and we finally obtain:

$$II \leq -\frac{1}{2\sigma_0^4} \sum_{j=1}^{p} \left(|\hat{\omega}_j|^2 - 1\right)^2 . \tag{19}$$

From Lemmas 6 and 7 and Equation (19), the following inequality is deduced from Equation (18), with probability at least $1 - \frac{3}{p}$:

$$\frac{1}{n}\left\|X\hat{G} - X\hat{G}_0\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 \leq C_1\sqrt{\frac{\log p}{n}}\sqrt{p+\hat{s}_0}\sqrt{\sum_{j=1}^{p}\left(|\hat{\omega}_j|^2-1\right)^2} - \frac{1}{2\sigma_0^4}\sum_{j=1}^{p}\left(|\hat{\omega}_j|^2-1\right)^2 + C_3\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1 + \lambda\left\|G_0\right\|_1 .$$

Let $\delta > 0$ such that $\delta \leq \frac{1}{2\sigma_0^4}$. Using $2xy \leq \frac{x^2}{a} + ay^2$ with $a = 2\delta$, we can show with probability at least $1 - \frac{3}{p}$, that:

$$\frac{1}{n}\left\|X\hat{G} - X\hat{G}_0\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 \leq \frac{C_1\log p}{4\delta n}(p+\hat{s}_0) + \left(\delta - \frac{1}{2\sigma_0^4}\right)\sum_{j=1}^{p}\left(|\hat{\omega}_j|^2-1\right)^2 + C_3\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1 + \lambda\left\|G_0\right\|_1 . \tag{20}$$

Lemma 8. *Assume that Assumptions* $\mathbf{H_{1-3}}$ *and* $\mathbf{H_{dim}}$ *hold. Then, with probability at least* $1 - \frac{2}{p}$*:*

$$\frac{1}{n}\left\|X\left(\hat{G}-\hat{G}_0\right)\right\|_F^2 \geq \left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2 \left\|\hat{G}-\hat{G}_0\right\|_F^2 .$$

Proof of Lemma 8. Denote $\|.\|_2$ the standard euclidian norm. This result is a consequence of Theorem 7.3 of van de Geer & Bühlmann (2013): for all $t > 0$, with probability at least $1 - 2e^{-t}$, we have:

$$\frac{1}{n}\|X\beta\|_2 \geq \left(\frac{3\lambda_{min}}{4} - \sqrt{\frac{2(t+\log p)}{n}} - 3\sigma_0\sqrt{\frac{s_\beta\log p}{n}}\right)\|\beta\|_2 , \tag{21}$$

uniformly on $\beta \in \mathbb{R}^p$, where s_β is the number of non-zero coefficients of β.

Moreover, $\frac{1}{n}\left\|X\left(\hat{G}-\hat{G}_0\right)\right\|_F^2 = \frac{1}{n}\sum_j \left\|X\left(\hat{G}-\hat{G}_0\right)^j\right\|_2^2$. Applying (21) to $\beta = \left(\hat{G}-\hat{G}_0\right)^j$ ($j \in [\![1,p]\!]$), with probability at least $1-2e^{-t}$:

$$\frac{1}{n}\left\|X\left(\hat{G}-\hat{G}_0\right)^j\right\|_2 \geq \left(\frac{3\lambda_{min}}{4} - \sqrt{\frac{2(t+\log p)}{n}} - 3\sigma_0\sqrt{\frac{s_{\left(\hat{G}-\hat{G}_0\right)^j}\log p}{n}}\right)\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_2,$$

where the quantity between brackets is non-negative by Assumption $\mathbf{H_2}$. The conclusion holds using $s_{\left(\hat{G}-\hat{G}_0\right)^j} \leq 2p$ and setting $t = \log p$. □

From Lemma 8 and Equation (20), we deduce that:

$$\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2\left\|\hat{G}-\hat{G}_0\right\|_F^2 - C_3\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1$$
$$+\left(\frac{1}{2\sigma_0^4}-\delta\right)\sum_{j=1}^{p}\left(|\hat{\omega}_j|^2-1\right)^2 + \lambda\left\|\hat{G}\right\|_1 \leq \frac{C_1\log p}{4\delta n}(p+\hat{s}_0) + \lambda\left\|G_0\right\|_1.$$

For all $j \in [\![1,p]\!]$, we finally use the Cauchy-Schwarz inequality:

$$\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_1 \leq \sqrt{s_{\left(\hat{G}-\hat{G}_0\right)^j}}\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_2 \leq \sqrt{2p}\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_2,$$

which gives:

$$\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1 \leq \sqrt{2}p\left\|\hat{G}-\hat{G}_0\right\|_F.$$

Therefore,

$$\underbrace{\left(\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2\left\|\hat{G}-\hat{G}_0\right\|_F - C_3p\sqrt{\frac{2\log p}{n}}\right)\left\|\hat{G}-\hat{G}_0\right\|_F}_{=A}$$
$$+\left(\frac{1}{2\sigma_0^4}-\delta\right)\sum_{j=1}^{p}\left(|\hat{\omega}_j|^2-1\right)^2 + \lambda\left\|\hat{G}\right\|_1$$
$$\leq \frac{C_1\log p}{4\delta n}(p+\hat{s}_0) + \lambda\left\|G_0\right\|_1 \leq \frac{C_1p^2\log p}{4\delta n} + \lambda s_{max}p\left\|G_0\right\|_\infty, \quad (22)$$

where we have used $\|G_0\|_1 = \sum_{i,j}|G_0|_i^j \leq \sum_{i=1}^{p} s_{max}\max_j|G_0|_i^j$.

Lemma 9 below gives us a bound for A.

Lemma 9.

$$\left(\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2\left\|\hat{G}-\hat{G}_0\right\|_F - C_3p\sqrt{\frac{2\log p}{n}}\right)\left\|\hat{G}-\hat{G}_0\right\|_F$$
$$\geq -\frac{C_3^2p^2\frac{\log p}{n}}{2\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}.$$

Proof of Lemma 9. A is minimal as soon as $\left\|\hat{G}-\hat{G}_0\right\|_F = \frac{Cp\sqrt{\frac{2\log p}{n}}}{2\left(\frac{3\lambda_{min}}{4}-2\sqrt{\frac{\log p}{n}}-3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}$, and its minimal value is:

$$A = -\frac{C_3^2 p^2 \frac{\log p}{n}}{2\left(\frac{3\lambda_{min}}{4}-2\sqrt{\frac{\log p}{n}}-3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}.$$

□

If η given in Assumption $\mathbf{H_{id}}$ satisfies

$$\eta \leq \frac{\frac{1}{2\sigma_0^4}-\delta}{\frac{C_1\log p}{4\delta n}p^2 + \lambda p s_{max}\left\|G_0\right\|_\infty + \frac{C_3^2 p^2\log p}{2n\left(\frac{3\lambda_{min}}{4}-2\sqrt{\frac{\log p}{n}}-3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}} \cdot p, \tag{23}$$

Equation (22) then contradicts Assumption $\mathbf{H_{id}}$. Setting $\lambda = C\sqrt{\frac{\log p}{n}s_{max}}$ with s_{max} chosen as in Assumption $\mathbf{H_3}$, η satisfies Equation (23) with probability at least $1-\frac{5}{p}$. We then deduce that the estimated permutation $\hat{P}$ is a good permutation, *i.e.* $\hat{P}\in\Pi_0$.

B.3 Inequalities in prediction and estimation

We now restart the proof with $\hat{P}\in\Pi_0$, $\hat{G}_0 = G_0$ and $\hat{\omega}_j = 1$, for all j. Equation (20) provides:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 \leq \frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1 + \lambda\left\|G_0\right\|_1. \tag{24}$$

We thus deduce that $\frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1 \leq \lambda\left(\left\|G_0\right\|_1 - \left\|\hat{G}\right\|_1 + \left\|\hat{G}-G_0\right\|_1\right)$. Denote $\mathcal{S}_0 = \{(i,j)\in [\![1,p]\!]^2,\ \ (G_0)_i^j \neq 0\}$ the support of G_0. Then, one has

$$\left\|G_0\right\|_1 - \left\|\hat{G}\right\|_1 + \left\|\hat{G}-G_0\right\|_1 \begin{cases} = 0 & \text{if } (i,j)\notin\mathcal{S}_0 \\ \leq 2\left|\left(\hat{G}-G_0\right)_i^j\right| & \text{otherwise,} \end{cases}$$

and we finally obtain:

$$\frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1 \leq 2\lambda\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1,$$

where $(\hat{G}-G_0)_{\mathcal{S}_0}$ is the matrix that has the same components as $\hat{G}-G_0$ on $\mathcal{S}_0$, 0 otherwise. Using $\left\|\hat{G}-G_0\right\|_1 = \left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 + \left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1$, where $\mathcal{S}_0^C$ is the notation for the complementary set of $\mathcal{S}_0$, the following inequality holds:

$$\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 \leq 3\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1. \tag{25}$$

We now apply Assumption $\mathbf{H_{RE}(s)}$ to the matrix $\hat{G}-G_0$ which satisfies Equation (25):

$$\kappa(s)\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_F \leq \frac{1}{\sqrt{n}}\left\|X\hat{G}-XG\right\|_F. \tag{26}$$

From Equation (24) and using the same calculus as previously, we can show that:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2+\frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1\leq 2\lambda\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1. \tag{27}$$

The Cauchy-Schwarz inequality now gives:

$$\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1\leq\sqrt{s_{max}}\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_F. \tag{28}$$

From Equations (26), (27) and (28), we finally obtain:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2+\frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1\leq\frac{2\lambda\sqrt{s_{max}}}{\kappa\left(s\right)\sqrt{n}}\left\|X\hat{G}-XG_0\right\|_F.$$

As a conclusion:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2\leq\frac{4\lambda^2 s_{max}}{\kappa^2\left(s\right)}.$$

The proof of the inequality in prediction follows with the definition of λ.

To obtain an inequality on estimation, remind that:

$$\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1\leq 3\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1.$$

We thus have

$$\begin{aligned}\left\|\hat{G}-G_0\right\|_1=\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1+\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 &\leq 4\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1\\ &\leq 4\sqrt{s_{max}}\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_F,\end{aligned}$$

with Cauchy-Schwarz inequality. Then, using again $\mathbf{H_{RE}(s)}$ to $\hat{G}-\hat{G}_0$ which satisfies Equation (24):

$$\begin{aligned}\left\|\hat{G}-G_0\right\|_1 &\leq \frac{4\sqrt{s_{max}}}{\kappa(s)}\frac{\left\|X\left(\hat{G}-G_0\right)\right\|_F}{\sqrt{n}}\\ &\leq \frac{16C}{\kappa^2(s)}\sqrt{\frac{\log p}{n}}s_{max}^{3/2},\end{aligned}$$

where we have used the inequality of prediction (7). This ends the proof.

References

Alander, J. (1992). On optimal population size of genetic algorithms. In *Proceedings of the IEEE Computer Systems and Software Engineering*.

Bach, F. (2008). Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the Twenty-fifth International Conference on Machine Learning*.

Banerjee, O., El Ghaoui, L. & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research* **9**, 485–516.

Barabási, A. & Oltvai, Z. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics* **5**, 101–113.

Bickel, P. & Li, B. (2006). Regularization in statistics. *Test* **15**, 271–344.

Bickel, P. J., Ritov, Y. & Tsybakov, A. B. (2009). Simultaneous analysis of lasso and Dantzig selector. *The Annals of Statistics* **37**, 1705–1732.

Breiman, L. (2001). Random forests. *Machine Learning* **45**, 532.

Bühlmann, P. (2013). Causal statistical inference in high dimensions. *Mathematical Methods of Operations Research* **77**, 357–370.

Bühlmann, P. & van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Publishing Company, Incorporated, 1st ed.

Cerf, R. (1998). Asymptotic convergence of genetic algorithms. *Advances in Applied Probability* **30**, 521–550.

Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. In *Learning from data (Fort Lauderdale, FL, 1995)*, vol. 112 of *Lecture Notes in Statist.* Springer, New York, pp. 121–130.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal Machine Learning Research* **3**, 507–554.

Cormen, T. H., Stein, C., Rivest, R. L. & Leiserson, C. E. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed.

Csiszár, I. & Tusnády, G. (1984). Information geometry and alternating minimization procedures. *Statistics & Decisions* **suppl. 1**, 205–237. Recent results in estimation theory and related topics.

Davis, L. et al. (1991). *Handbook of genetic algorithms*, vol. 115. Van Nostrand Reinhold New York.

de la Fuente, A., ed. (2013). *Gene Network Inference, Verification of Methods for Systems Genetics Data*, chap. A Panel of Learning Methods for the Reconstruction of Gene Regulatory Networks in a Systems Genetics. Springer, pp. 9–31.

de la Fuente, A. & Stolovitzky, G. (2010). Dream5 systems genetics challenge. `http://wiki.c2b2.columbia.edu/dream/index.php/D5c3`.

Dréo, J. (2006). *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media.

Duchi, J., Gould, S. & Koller, D. (2008). Projected subgradient methods for learning sparse gaussians. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics* **32**, 407–499.

El Karoui, N. (2008). Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics* **36**, 2757–2790.

Ellis, B. & Wong, W. (2008). Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association* **103**, 778–789.

Evtushenko, Y., Malkova, V. & Stanevichyus, A. (2009). Parallel global optimization of functions of several variables. *Comput. Math. Math. Phys.* **49**, 246–260.

Friedman, J., Hastie, T. & Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**, 432–441.

Friedman, J., Hastie, T. & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

Friedman, N. & Koller, D. (2003). Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* **50**, 95–125.

Fu, F. & Zhou, Q. (2013). Learning sparse causal Gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association* **108**, 288–300.

Giraud, C. (2014). *Introduction to High-Dimensional Statistics*. Monographs on Statistics & Applied Probability. CRC Press.

Granville, V., Krivanek, M. & Rasson, J.-P. (1994). Simulated annealing: A proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 652–656.

Grefenstette, J., Gopal, R., Rosmaita, B. & Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications*.

Grzegorczyk, M. & Husmeier, D. (2008). Improving the structure MCMC sampler for bayesian networks by introducing a new edge reversal move. *Machine Learning* **71**.

Guyon, I., Aliferis, C. & Cooper, G., eds. (2010). *Causation and Prediction Challenge: Challenges in Machine Learning*, vol. 2. Microtome Publishing.

Hauser, A. & Bühlmann, P. (2012). Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research* **13**, 2409–2464.

Hauser, A. & Bühlmann, P. (2015). Jointly interventional and observational data: estimation of interventional Markov equivalence classes of directed acyclic graphs. *Journal of the Royal Statistical Society. Series B* **77**, 291–318.

Hiriart-Urruty, J. B. & Lemaréchal, C. (1993). *Convex analysis and minimization algorithms. II*, vol. 306 of *Fundamental Principles of Mathematical Sciences*. Berlin: Springer-Verlag.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press.

Horst, R. & Pardalos, P., eds. (1995). *Handbook of global optimization*. Nonconvex optimization and its applications. Dordrecht, Boston: Kluwer Academic Publishers.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. & Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE* **5**, e12776.

Jones, D. R., Perttunen, C. D. & Stuckman, B. E. (1993). Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications* **79**, 157–181.

Kahn, A. (1962). Topological sorting of large networks. *Communications of the ACM* **5**, 558562.

Kalisch, M. & Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research* **8**, 613–636.

Koivisto, M. & Sood, K. (2004). Exact bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research* **5**, 549–573.

Ledoit, O. & Wolf, M. (2015). Spectrum estimation: A unified framework for covariance matrix estimation and pca in large dimensions. *Journal of Multivariate Analysis* **139**, 360384.

Liaw, A. & Wiener, M. (2002). Classification and regression by randomforest. *R News* **2**, 18–22.

Liu, H., Wang, L. & Zhao, T. (2014). Sparse covariance matrix estimation with eigenvalue constraints. *Journal of Computational and Graphical Statistics* **23**, 439–459.

Lounici, K., Pontil, M., Tsybakov, A. & van de Geer, S. (2009). Taking advantage of sparsity in multi-task learning. In *Proceedings of the 22nd Conference on Learning Theory*.

Marbach, D., Schaffter, T. & Floreano, D. (2009a). Dream4 in silico network challenge. `http://wiki.c2b2.columbia.edu/dream/index.php?title=D4c2`.

Marbach, D., Schaffter, T., Mattiussi, C. & Floreano, D. (2009b). Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology* **16**, 229–239.

Meinshausen, N. & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics* **34**, 1436–1462.

Mestre, X. (2008). Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates. *IEEE Transactions on Information Theory* **54**, 5113–5129.

Michalewicz, Z. (1994). *Genetic algorithms + data structures = evolution programs*. Berlin: Springer-Verlag, 2nd ed.

Newman, M. (2003). The structure and function of complex networks. *SIAM Review* **45**, 157–256.

Pearl, J. (2009). *Causality*. Cambridge: Cambridge University Press. Models, reasoning, and inference.

Peters, J., Mooij, J., Janzing, D. & Schölkopf, B. (2014). Causal discovery with continuous additive noise models. *Journal of Machine Learning Research* **15**, 2009–2053.

Peters, J., Mooij, J. M., Janzing, D. & Schölkopf, B. (2011). Identifiability of causal graphs using functional models. In *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*.

Petersen, K. B. & Pedersen, M. S. (2012). The matrix cookbook. `http://www2.imm.dtu.dk/pubdb/p.php?3274`. Version 20121115.

Piszcz, A. & Soule, T. (2006). Genetic programming: Optimal population sizes for varying complexity problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

Rau, A., Jaffr'ezic, F. & Nuel, G. (2013). Joint estimation of causal effects from observational and intervention gene expression data. *BMC Systems Biology* **7**, 111.

Ridge, E. (2007). *Design of Experiments for the Tuning of Optimisation Algorithm*. Ph.D. thesis, The University of York, Department of Computer Science.

Schaffer, J. D., Caruana, R., Eshelman, L. J. & Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*.

Schwarz, G. E. (1978). Estimating the dimension of a model. *The Annals of Statistics* **6**, 461–464.

Sergeyev, Y. (1995). An information global optimization algorithm with local tuning. *SIAM Journal on Optimization* **5**, 858–870.

Sergeyev, Y. & Kvasov, D. (2006). Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM Journal on Optimization* **16**, 910–937.

Shojaie, A. & Michailidis, G. (2010). Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika* **97**, 519–538.

Silander, T. & Myllymäki, T. (2006). A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence (UAI)*.

Souma, W., Fujiwara, Y. & Aoyama, H. (2006). *The Complex Networks of Economic Interactions*, chap. Heterogeneous Economic Networks. Springer, pp. 79–92.

Spirtes, P., Glymour, C. & Scheines, R. (2000). *Causation, prediction, and search*. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2nd ed. With additional material by David Heckerman, Christopher Meek, Gregory F. Cooper and Thomas Richardson, A Bradford Book.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B* **58**, 267–288.

Tsarmadinos, I., Brown, L. & Aliferis, C. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* **65**, 31–78.

van de Geer, S. & Bühlmann, P. (2013). ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics* **41**, 536–567.

van de Geer, S., Bühlmann, P. & Zhou, S. (2011). The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics* **5**, 688–749.

Verma, T., Araújo, N. & Herrmann, H. (2014). Revealing the structure of the world airline network. *Scientific Reports* **5**.

Verma, T. & Pearl, J. (1991). Equivalence and synthesis of causal models. In *Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*.

Verzelen, N. (2012). Minimax risks for sparse regressions: Ultra-high dimensional phenomenons. *Electronic Journal of Statistics* **6**, 38–90.

Vignes, M., Vandel, J., Allouche, D., Ramadan-Alban, N., Cierco-Ayrolles, C., Schiex, T., Mangin, B. & de Givry, S. (2011). Gene regulatory network reconstruction using bayesian networks, the dantzig selector, the lasso and their meta-analysis. *PLoS ONE* **6**, e29165.

Wainwright, M. (2009). Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory* **55**, 5728–5741.

Witten, D., Freidman, J. & Simon, N. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics* **20**, 892–900.

Yuan, M. & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika* **94**, 19–35.

Zhou, Q. (2011). Multi-domain sampling with applications to structural inference of Bayesian networks. *Journal of the American Statistical Association* **106**, 1317–1330.