Convergence of Feedback Arc Set-Based Heuristics for Linear Structural Equation Models

Pierre Gillot Pierre.Gillot@uib.no

Pekka Parviainen Pekka.Parviainen@uib.no

University of Bergen HIB - Thormøhlens gate 55 Postboks 7803 5020 Bergen

Abstract

Score-based structure learning in Bayesian networks, where local structures in the graph are given a score and one seeks to recover a high-scoring DAG from data, is an NP-hard problem. While the general learning problem is combinatorial, the more restricted framework of linear structural equation models (SEMs) enables learning Bayesian networks using continuous optimization methods. Large scale structure learning has become an important problem in linear SEMs and many approximate methods have been developed to address it. Among them, feedback arc set-based methods learn the DAG by alternating between unconstrained gradient descent-based step to optimize an objective function and solving a maximum acyclic subgraph problem to enforce acyclicity. In the present work, we build upon previous contributions on such heuristics by first establishing mathematical convergence analysis, previously lacking; second, we show empirically how one can significantly speed-up convergence in practice using simple warmstarting strategies.

Keywords: Bayesian networks; Structure learning; Linear structural equation models; Convex optimization; Maximum acyclic subgraph.

1. Introduction

Bayesian networks are a class of probabilistic graphical models where the conditional independencies between variables are expressed using a directed acyclic graph (DAG). We are interested in the structure learning problem, that is, how to construct the DAG based on data. We take a score-based approach to structure learning where every DAG is assigned a score based on how well it fits to the data and one tries to find a DAG that optimizes the score. Typically, the score decomposes into a sum of local scores that are computed for node-parent set pairs. This yields a combinatorial optimization problem where one picks a parent set for each node and tries to maximize the sum of the local scores while constraining the resulting graph to be acyclic. This problem is known to be NP-hard (Chickering, 1996).

In this paper, we concentrate on linear structural equation models (linear SEMs) which are a subclass of Bayesian networks. They are used to model continuous variables and the value of a variable depends linearly on values of its parents. From the learning perspective, linear SEMs simplify the optimization because the score function depends only on the arc weights and not the node-parent set pairs. However, the structure learning problem remains combinatorial due to the acyclicity constraint imposed to the graph.

Recently, Zheng et al. (2018) introduced a continuous acyclicity constraint that enables learning SEMs with continuous optimization instead of combinatorial optimization. However, learning DAGs using the acyclicity function proposed in Zheng et al. (2018) proves impractical in large scale settings owing to the complexity of the matrix exponential, which

exhibits cubic time complexity and quadratic space complexity with respect to the number of nodes. Various methods have been recently developed in order to circumvent this problem and enable learning large structures. As a rule, these methods avoid encoding acyclicity with hard constraints and instead formulate alternative problems that can be solved with lower complexity per iteration. We note however that these new problems remain largely non-convex in nature and one cannot hope to find a global minimizer in general. In Yu et al. (2021), a cyclic solution is first computed and then projected to the DAG space using a novel characterization based on Hodge decomposition of graphs. In Zhu et al. (2021), the hard constraint encoded by the acyclicity function is relaxed and an upped-bound on the spectral radius of a non-negative adjacency matrix of the graph is derived instead. In Dong and Sebag (2022), low-rank solutions are combined with an efficient approximation for the computation of the gradient of the acyclicity function.

Alternatively, combining feedback arc set heuristics with continuous optimization schemes has proved successful in learning large scale DAGs. Such methods consist in decoupling the optimization of the objective function from acyclicity itself by alternating between fast gradient-based optimization steps without acyclicity and projection of cyclic solutions to "close" acyclic approximations; while Park and Klabjan (2017) greedily fit parameters of a newly discovered acyclic structure at every step, instead Gillot and Parviainen (2022) dynamically construct a sequence of convex objective functions penalized to remain in the vicinity of a trail of acyclic solutions discovered online, resulting in better scalability but losing theoretical guarantees on the convergence of their method.

The present paper has two contributions. The first contribution is theoretical. We show that the *ProxiMAS* algorithm presented in Gillot and Parviainen (2022) converges under certain conditions (Lemma 2, Theorem 3). We note that the conditions are stronger than for the *GD* algorithm by Park and Klabjan (2017). Second, we analyse the convergence of ProxiMAS empirically. We also show that clever warmstarting strategies can lead to substantially faster convergence for feedback arc set heuristic-based structure learning.

2. Background

2.1 Linear Structural Equation Models and Bayesian Network Structure Learning

Let V be a node set. Furthermore, let $G = (V, A)$ be a DAG where A is the arc set. The parent set of node v in G is denoted by A_v.

Formally, a Bayesian network is a pair (G, Θ) where its structure G is a DAG and Θ are its parameters. The joint distribution factorizes as follows:

$$P(V) = \prod_{v \in V} P(v|A_v, \theta_v)$$

where θ_v are parameters of the conditional distribution of v given its parents.

Linear SEMs are a special case of Bayesian networks. To specify a model, we have a weight matrix $W \in \mathbb{R}^{d \times d}$, where d is the cardinality of the node set V. The weight matrix specifies both the structure and parameters of the Bayesian network. Specifically, $W(i, j) \neq 0$ entails there is an arc going from i to j in the DAG. Given a d-dimensional

data vector x, a linear SEM can be written as

$$x = xW + \epsilon$$

where ϵ is a d-dimensional error vector. The elements of ϵ are independent. The data consists of n such samples x, forming a data matrix $X \in \mathbb{R}^{n \times d}$. We note that when the errors are Gaussian, linear SEMs encode multivariate Gaussian distributions.

To learn a linear SEM, we need to constrain W to represent an acyclic graph. Furthermore, one typically uses the least squares loss and adds a regularization term inducing sparsity in the structure. Thus, the objective function in structure learning becomes

$$\underset{W}{\operatorname{argmin}} \frac{1}{2n}||XW - X||^2 + \lambda g(W) \quad \text{s.t. } W \text{ is acyclic} \tag{1}$$

where $\|\cdot\|$ is the Frobenius norm and $g(W)$ is for regularization, whose strength is controlled by the hyperparameter $\lambda > 0$.

2.2 Feedback Arc Set-Based Structure Learning

At a general level, feedback arc set-based methods learn a DAG (under the linear SEMs framework) by iteratively repeating the following steps:

- Given an acyclic graph, find a graph (possibly cyclic) which is better in terms of the objective function value.
- Given a cyclic graph, find a close acyclic graph by approximately solving a maximum acyclic subgraph instance.

In particular, these methods entirely decouple acyclicity from the optimization process itself, via the integration of (weighted) maximum acyclic subgraph (MAS) problems, whose definition we recall now: given a directed graph $G = (V, E)$ and a weight function $w(e)$ that assigns a weight for each arc $e \in E$, the goal is to find an acyclic graph $G' = (V, E')$ such that $E' \subset E$ and $\sum_{e \in E'} w(e)$ is maximized. The dual problem is called the feedback arc set (FAS) problem: given a directed graph $G = (V, E)$ and a weight function $w(e)$, the goal is to find an arc set $E'' \subset E$ such that $G'' = (V, E \setminus E'')$ is acyclic and $\sum_{e \in E''} w(e)$ is minimized. Given a cyclic graph G as input, it is well known that G' is an optimal solution of MAS if and only if $G \setminus G'$ is an optimal solution of FAS. Moreover, both problems are NP-hard (Karp, 1972).

Intuitively, using the maximum acyclic subgraph problem in order to learn linear SEMs DAGs is sensible, in that given any acyclic solution to linear SEMs, one can always extend this solution into a tournament (a dense acyclic graph) having the exact same score, by completing the solution with zero-weight arcs. Unlike traditional approaches that involve a smooth characterization of acyclicity (Zheng et al., 2018; Ng et al., 2020; Yu et al., 2021; Zhu et al., 2021; Dong and Sebag, 2022), feedback arc set-based methods also offer the clear advantage that they return strictly acyclic solutions, in the sense that one never needs to threshold a solution as a form of postprocessing in order to recover a DAG.

Two variants have been studied so far. In Park and Klabjan (2017), the authors propose the GD algorithm which works by repeating the following steps:

1. Fix the structure of the last obtained acyclic solution, then fit the linear SEMs objective constrained by this structure to get a new fitted acyclic solution.

2. Make an unconstrained optimization step on the linear SEMs loss at the previously obtained fitted acyclic solution to get a new cyclic solution.

3. Project the previously obtained cyclic solution to its maximum acyclic subgraph approximation to get a new acyclic solution.

The key design choice in GD lies in the fact that unconstrained optimization steps are only performed after a newly found structure has been fitted with respect to the linear SEMs objective. From a theoretical perspective, this leads to a simplified convergence analysis and GD is guaranteed to converge in a fix number of iterations under mild conditions (see (Park and Klabjan, 2017), Lemma 1). On the practical side however, the GD algorithm "greedily" explores the search space which can lead to overfitting and incurs solving a LASSO subproblem for every node in the graph at every iteration, heavily impacting the scalability of the algorithm. In Gillot and Parviainen (2022), an alternative approach is proposed that would fix the scalability concern observed in GD. This new variant changes steps 1 and 2 from GD (step 3 is left unchanged) as follows:

1'. Construct a new objective function as the sum of the linear SEMs loss plus a least-squared term penalizing deviation from the last obtained acyclic solution.

2'. Make an unconstrained optimization step on the previously constructed objective function to get a new cyclic solution.

In other words, this second approach jumps from an acyclic structure to another, without fitting these structures to optimality. As a trade-off, the optimization process now evolves dynamically, making a convergence analysis less straightforward (and such analysis is presently missing, to the best of our knowledge). The pseudocode of this variant is described in Algorithm 1.

Algorithm 1 (Gillot and Parviainen, 2022)

Input: $X \in \mathbb{R}^{n \times d}$, $\lambda > 0$, $\mu > 0$

$\widetilde{W}_0, W_0 = 0^{d \times d}$

for $1 \leq k \leq \dots$ **do**

  New objective function: $\phi_k \colon W \mapsto \frac{1}{2n}\|XW - X\|^2 + \frac{\mu}{2}\|W - W_{k-1}\|^2 + \lambda\|W\|_1$

  Optimization step: $\widetilde{W}_k = \text{step}(\phi_k, \text{ optimizer})$

  MAS projection: $W_k = \text{MAS}(\widetilde{W}_k)$

In short, the algorithm keeps track of both cyclic and acyclic solutions, represented respectively by $\widetilde{W}_k$ and W_k. At every iteration, a new objective function is constructed: let $f \colon W \mapsto \frac{1}{2n}\|XW - X\|^2$ and $g \colon W \mapsto \lambda\|W\|_1$ represent the linear SEMs loss and the sparsity inducing penalization term respectively; let $f_k : W \mapsto f(W) + \frac{\mu}{2}\|W - W_{k-1}\|^2$ represent the linear SEMs loss penalized to remain in the vicinity of the previously discovered acyclic structure; then the new objective function is $\phi_k = f_k + g$, where both f_k and g are convex,

the f_k are differentiable and their gradient share the same optimal Lipschitz constant $L = \frac{1}{n}\|X^tX + n\mu I_d\|_*$ (where $\|\cdot\|_*$ is the spectral norm). From a practical standpoint, this means that one can exploit the stationary properties of the ϕ_k in order to make fast progress with a proximal gradient-based optimizer, though Algorithm 1 can embed instead any gradient-based first-order optimizer. The authors dub the former *ProxiMAS* and the latter *OptiMAS*. We note that while the objective functions ϕ_k are all convex, the overall optimization scheme itself remains largely non-convex, in that every function ϕ_k carries structural information which can evolve in a non-convex fashion from an iteration to another. This structural information is encapsulated within the acyclic solutions W_k. In order to construct them a feedback arc set heuristic is used, which at every iteration k constructs a topological order π_k from the cyclic solution $\widetilde{W}_k$; the acyclic projection W_k is obtained by nullifying those weights in $\widetilde{W}_k$ corresponding to feedback arc set arcs (with respect to π_k). Both Park and Klabjan (2017) and Gillot and Parviainen (2022) make use of a variant of the greedy feedback arc set heuristic originally presented in Eades et al. (1993). More specifically, this variant iteratively constructs a topological order from its last/rightmost up to its first/leftmost element. A node is greedily selected if it has the smallest sum of incoming squared weights among the remaining nodes. In other words, this heuristic treats forward arcs (with respect to the constructed topological order) as feedback arc set arcs. It is described in Algorithm 2.

Algorithm 2 Greedy feedback arc set heuristic

Input: $\widetilde{W} \in \mathbb{R}^{d\times d}$

$V_1 = \{0, \dots, d-1\}, \pi = 0^{d\times 1}$
for $1 \leq r \leq d$ **do**
  $\pi[-r] = \operatorname*{argmin}_{j\in V_r} \left\|\widetilde{W}[:, j]\right\|^2_{V_r\setminus\{j\}}$
  $V_{r+1} = V_r \setminus \pi[-r]$
return π

3. Convergence Analysis

Minimizing a composite convex function is a standard problem in convex analysis: let $\phi \coloneqq f + g : U \mapsto \mathbb{R}$ denote a composite convex function on a convex open set $U \subset \mathbb{R}^m$ such that: f and g are convex on U, f is differentiable and its gradient is Lipschitz-continuous with constant L on U. Then it is well known that the non-accelerated proximal gradient descent optimizer generating the sequence $(x_k)_k$ defined as

$$x_k = \operatorname*{argmin}_{x\in U} \left\{ \frac{\gamma_k^{-1}}{2}\left\|x - \left(x_{k-1} - \gamma_k \nabla f(x_{k-1})\right)\right\|^2 + g(x)\right\} \quad \text{where } 0 < \gamma_k \leq L^{-1} \tag{2}$$

achieves $\mathcal{O}\left(\frac{1}{k}\right)$ convergence rate in function value (where k is the number of iterations) (Beck and Teboulle, 2009a). A key aspect of the convergence analysis is to show that one in fact always has (see for instance (Beck and Teboulle, 2009a), Lemma 1.6):

$$0 < \gamma_k \leq L^{-1} \implies \phi(x_k) \leq \phi(x_{k-1}) - \frac{\gamma_k^{-1}}{2}\|x_k - x_{k-1}\|^2, \tag{3}$$

that is the proximal gradient descent update generates a sequence guaranteed to decrease the objective function value. Algorithm 1 equipped with the same convex optimizer subtly

differs from this framework, in that at every iteration a convex descent step is performed on a new composite convex function $\phi_k = f_k + g$: this describes a dynamic system and the notion of optimal solution is ill-defined, thus the $\mathcal{O}\left(\frac{1}{k}\right)$ convergence rate in function value is lost. In the rest of this section, by ProxiMAS we refer to Algorithm 1 equipped with both Algorithm 2 for the FAS heuristic and the non-accelerated proximal gradient descent optimizer described above, i.e. ProxiMAS makes convex descent steps of the form:

$$\widetilde{W}_k = \underset{W \in \mathbb{R}^{d\times d}}{\operatorname{argmin}} \left\{ \frac{\gamma_k^{-1}}{2} \left\| W - \left(\widetilde{W}_{k-1} - \gamma_k \nabla f_k\left(\widetilde{W}_{k-1}\right)\right)\right\|^2 + \lambda \|W\|_1 \right\} \quad \text{where } 0 < \gamma_k \leq L^{-1}, \tag{4}$$

where all f_k have Lipschitz-continuous gradient with the same constant L. We aim to derive a set of conditions such that ProxiMAS converges to a fixed acyclic structure in a finite number of iterations, that is the acyclic solutions W_k have the same support, or equivalently the topological orders π_k constructed by Algorithm 2 are the same. Lemma 1 provides a necessary condition, agnostic from the choice of the optimizer (in Algorithm 1: line 4):

Lemma 1 *Let $\left(\widetilde{W}_k\right)_k$, $(W_k)_k$ and $(\pi_k)_k$ respectively denote the sequence of cyclic solutions, acyclic solutions and topological orders in Algorithm 1. Assume topological orders stabilize, i.e. $\exists k_1 : \forall k \geq k_1,\ \pi_k = \pi$. Then the following convergence condition necessarily holds:*

$$\exists k_0 : \forall k \geq k_0,\ \left\|\widetilde{W}_k - W_k\right\| \leq \left\|\widetilde{W}_k - W_{k-1}\right\|. \tag{5}$$

Proof Notice that $\left\|\widetilde{W}_k - W_k\right\|^2 \leq \left\|\widetilde{W}_k - W_{k-1}\right\|^2 \iff \left\|W_k - W_{k-1}\right\|^2 \geq 2\left\langle \widetilde{W}_k - W_k, W_{k-1}\right\rangle$. Assuming that for large k, $\pi_k = \pi$, one must then have $\left\langle \widetilde{W}_k - W_k, W_{k-1}\right\rangle = 0$. Indeed, non-zero values in $\widetilde{W}_k - W_k$ must correspond to forward arcs whereas non-zero values in W_{k-1} must correspond to backward arcs (both with respect to π for large enough k). ■

In order to get the convergence of acyclic solutions W_k, we must first ensure we get the convergence of cyclic solutions $\widetilde{W}_k$. We stress that by convergence we imply toward a local extremum and that converging does not guarantee good performance of found solutions, that is we are concerned with the stability of ProxiMAS. We prove the following:

Lemma 2 *Let $\left(\widetilde{W}_k\right)_k$, $(W_k)_k$ and $(\gamma_k)_k$ respectively denote the sequence of cyclic solutions, acyclic solutions and learning rates in ProxiMAS. Assume the learning rate decreases with rate $\mathcal{O}\left(\frac{1}{k^\alpha}\right)$ where $\alpha > 2$, and assume the convergence condition from Lemma 1 holds: $\exists k_0 : \forall k \geq k_0,\ \left\|\widetilde{W}_k - W_k\right\| \leq \left\|\widetilde{W}_k - W_{k-1}\right\|$. Then $\widetilde{W}_k$ admits a convergent subsequence.*

Proof Notice the ϕ_k have stationary properties (composite convex functions, same optimal Lipschitz constant L for the gradient of smooth components) hence Equation 3 holds for every ϕ_k at step k: $\forall k \geq 1,\ 0 < \gamma_k \leq L^{-1} \implies \phi_k\left(\widetilde{W}_k\right) \leq \phi_k\left(\widetilde{W}_{k-1}\right) - \frac{\gamma_k^{-1}}{2}\left\|\widetilde{W}_k - \widetilde{W}_{k-1}\right\|^2$. Now, by definition: $\phi_k\left(\widetilde{W}_{k-1}\right) = \phi_{k-1}\left(\widetilde{W}_{k-1}\right) + \frac{\mu}{2}\left(\left\|\widetilde{W}_{k-1} - W_{k-1}\right\|^2 - \left\|\widetilde{W}_{k-1} - W_{k-2}\right\|^2\right)$. Due to the convergence condition, we thus get $\phi_k\left(\widetilde{W}_k\right) \leq \phi_k\left(\widetilde{W}_{k-1}\right) \leq \phi_{k-1}\left(\widetilde{W}_{k-1}\right)$ for large k, implying the (non-negative) sequence $\left(\phi_k\left(\widetilde{W}_k\right)\right)_k$ converges to a limit l. Furthermore, we can now write that for large k, $\frac{\gamma_k^{-1}}{2}\left\|\widetilde{W}_k - \widetilde{W}_{k-1}\right\|^2 \leq \phi_{k-1}\left(\widetilde{W}_{k-1}\right) - \phi_k\left(\widetilde{W}_k\right)$. We then use the fact that the right-hand side in the previous inequality is a telescopic term, along with $\phi_k\left(\widetilde{W}_k\right) \underset{k\to+\infty}{\longrightarrow} l$, to deduce that the infinite series $\sum_k \gamma_k^{-1}\left\|\widetilde{W}_k - \widetilde{W}_{k-1}\right\|^2$ converges;

necessarily, $\gamma_k^{-1}\|\widetilde{W}_k - \widetilde{W}_{k-1}\|^2 = o(1)$ [1] holds, which in turn implies $\|\widetilde{W}_k - \widetilde{W}_{k-1}\| = \mathcal{O}(\sqrt{\gamma_k})$. Now by assumption $\sqrt{\gamma_k} = \mathcal{O}(\frac{1}{k^{\alpha/2}})$ where $\alpha > 2$, hence $\|\widetilde{W}_k - \widetilde{W}_{k-1}\| = \mathcal{O}(\frac{1}{k^{\beta}})$ where $\beta > 1$ such that the infinite series $S \coloneqq \sum_k \|\widetilde{W}_k - \widetilde{W}_{k-1}\|$ converges. The triangular inequality finally yields:

$$\forall K,\ \|\widetilde{W}_K - \widetilde{W}_0\| = \|\textstyle\sum_{k\leq K}\widetilde{W}_k - \widetilde{W}_{k-1}\| \leq \sum_{k\leq K}\|\widetilde{W}_k - \widetilde{W}_{k-1}\| \leq S < +\infty,$$

therefore $\sup_k \|\widetilde{W}_k\| < +\infty$. The Bolzano-Weierstrass theorem concludes the proof. ■

We are now ready to present our main result:

Theorem 3 *Let $(\widetilde{W}_k)_k$ and $(\pi_k)_k$ respectively denote the sequence of cyclic solutions and topological orders in ProxiMAS. Assume $(\widetilde{W}_k)_k$ admits a converging subsequence: $\widetilde{W}_* \coloneqq \lim_{k\to+\infty}(\widetilde{W}_{\psi(k)})_k$. Define π_* to be the topological order constructed by Algorithm 2 given $\widetilde{W}_*$ as input and assume for all $r \in [1, d]$, Algorithm 2 makes a strictly optimal decision when constructing $\pi_*[-r]$ (i.e. argmin in Algorithm 2: line 3 is strict at every step r given $\widetilde{W}_*$ as input). Then the topological orders constructed by ProxiMAS in the subsequence ψ stabilize after a finite number of iterations: $\exists k' : \forall k \geq k',\ \pi_{\psi(k)} = \pi_*$.*

Proof idea The proof is technical and revolves around a similar argument as in Park and Klabjan (2017): Lemma 1. Due to space constraints, we leave out the full proof. ■

We note that the assumption in Theorem 3 is mild: although one never has access to the limit of a converging subsequence, arc weights are continuous thus Algorithm 2 easily makes strictly optimal choices. However, columns of zeros can occur in practice (e.g. when learning sparse structures), in which case convergence cannot be guaranteed. We also comment on Lemma 2's assumptions: the convergence condition from Lemma 1 ensures feedback arc set costs eventually become less than the distance between past acyclic solutions and new cyclic solutions; the learning rate must decrease sufficiently fast which can deteriorate the quality of found solutions. These two assumptions are not needed in the theoretical convergence of GD (Park and Klabjan, 2017), meaning the theoretical convergence of ProxiMAS (Gillot and Parviainen, 2022) is weaker. This was expected since unlike GD, ProxiMAS does not solve LASSO subproblems at every iteration.

4. Experiments

We now conduct an empirical study of feedback arc set-based heuristics for linear SEMs. This study is divided into three experiments. First, we empirically validate the convergence analysis of ProxiMAS by investigating the stability of the method in various settings; second, we assess the influence of the MAS penalization hyperparameter μ with different convex optimizers in ProxiMAS; third, we compare different warmstarting strategies in order to speed-up the practical convergence of feedback arc set-based heuristics.

1. If $k\gamma_k^{-1}\|\widetilde{W}_k - \widetilde{W}_{k-1}\|^2$ has a limit in $\mathbb{R}_+ \cup \{+\infty\}$, one in fact has $\gamma_k^{-1}\|\widetilde{W}_k - \widetilde{W}_{k-1}\|^2 = o(\frac{1}{k})$ (due to the divergence of the harmonic series); in that case $\gamma_k = \mathcal{O}(\frac{1}{k^{\alpha}})$ where $\alpha > 1$ suffices for Lemma 2 to hold.

4.1 Setup

We consider a setup similar to that found in Zheng et al. (2018). Data generation is as follows: we start by generating an undirected graph with d nodes from two classes of random graphs, namely Erdős-Rényi ("ER") and scale-free ("SF"). Assuming the graph is sampled to have average degree δ, we refer to this graph as "ERδ" (respectively "SFδ"). A random permutation is then sampled and assigned to the graph which yields a DAG. Next, arc weights W are uniformly sampled in the range $[-2,-0.5] \cup [0.5,2]$. The last step is to generate linear SEMs samples $X = \mathcal{E}(I-W)^{-1}$, where $\mathcal{E} \in \mathbb{R}^{n \times d}$ represents n noise samples, with $n = 0.1 \times d$ (low sample count) or $n = 10 \times d$ (large sample count). We restrict $\mathcal{E}$ to be generated from Gaussian noise only and study both the equal variance setting ("EV": all σ equal 1.0) and the non-equal variance setting ("NV": all σ uniformly sampled in $[0.5, 1.5]$). In all considered experiments, 20 instances are randomly generated as described above; we represent variance in our figures with shaded regions.

We always fix the sparsity-inducing hyperparameter λ to 0.1, as in Gillot and Parviainen (2022). The number of iterations allowed for tested methods is always set to ten times the number of nodes (e.g. 10000 iterations when $d = 1000$). Every 100 iterations a snapshot is recorded and different metrics are extracted, such as the loss (Equation 1) of the current acyclic solution and its average precision with respect to the true DAG (Markov equivalence is ignored). We consider as well metrics to assess the convergence of tested heuristics, such as: the order matching metric which gives the percentage of matching nodes in two consecutive topological orders constructed by Algorithm 2; the convergence condition metric which evaluates the quantities $\left\|\widetilde{W}_k - W_k\right\| - \left\|\widetilde{W}_k - W_{k-1}\right\|$ (remember that these quantities must remain negative after a finite number of iterations to guarantee structural convergence, see Lemma 1). Both the order matching and the convergence condition metrics are averaged over the past 100 iterations to get smoother estimates. Implementation is based on pytorch 1.10 and experiments were run on a cluster with Intel Xeon-Gold 6138 2.0 GHz / 6230R 2.1 GHz CPU cores. Table 1 lists the hyperparameters for each experiment. To save space we only show a subset of all figures.

Exp	$\boldsymbol{d}$	$\boldsymbol{\delta}$	$\boldsymbol{\mu}$	**Convex opti**	**Const lr %**	**Cyclic %**	**Convex %**
1	1000	$1, 2, 4$	10^{δ}	I, F	$0, 50, 100$	0	100
2	1000	4	$10^{i}, 1 \leq i \leq \delta$	I, F, G, N	100	0	100
3	2000	$4, 8$	10^{δ}	F	100	$0, 50$	$0, 20, \ldots, 100$

Table 1: Experiments hyperparameters (I: ISTA; F: FISTA; G: Greedy FISTA; N: Nesterov)

4.2 Experiment 1

In the first experiment we consider the classical iterative shrinkage-thresholding algorithm (ISTA) implementing in closed-form Equation 2 and its well known accelerated variant FISTA (Beck and Teboulle, 2009b), with different learning rate strategies. The learning rate is implemented to decrease with rate $\frac{1}{k^{1.001}}$, but remains constant for $x\%$ of the total number of iterations before decreasing (x varies as described in Table 1: column "Const lr %"). Fig-

ure 1 illustrates Experiment 1. Looking at the average precision curves, clearly ProxiMAS stabilizes once the learning rate starts decreasing (orange curves). When applied too early, decrease in learning rate hurts performance (pink curves). Comparing the optimizers, ISTA is a much slower learner than FISTA and fails to learn denser graphs ($\delta = 4$). Looking at the convergence condition metric, a decreasing learning rate yields infinitesimal quantities $\left\|\widetilde{W}_k - W_k\right\| - \left\|\widetilde{W}_k - W_{k-1}\right\|$ for ISTA. With constant learning rate (teal curves) FISTA learns fast and eventually satisfies the convergence condition $\left\|\widetilde{W}_k - W_k\right\| \leq \left\|\widetilde{W}_k - W_{k-1}\right\|$. Experiment 1 suggests that ProxiMAS is stable in practice: even when the convex optimizer is accelerated, the learning rate is constant and the convergence condition from Lemma 1 does not exactly hold, ProxiMAS reaches a performance plateau.

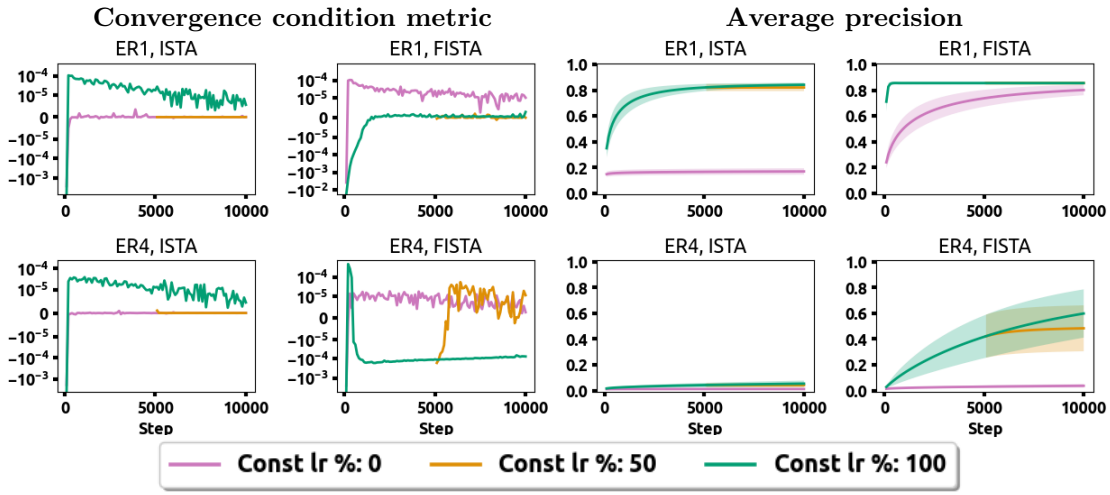

Figure 1: Experiment 1: learning rate policy varies ($d = 1000$, $n = 10000$, NV).

4.3 Experiment 2

In the second experiment we compare the behaviour of various convex optimizers with respect to the hyperparameter μ controlling the strength of the MAS penalization terms. In addition to the classical ISTA and FISTA optimizers, we consider the Greedy FISTA variant described in Liang et al. (2022) that relies on restarting. We consider as well the Nesterov variant outlined in Nesterov (2014) (refer to "Constant Step Scheme, III") which unlike aforementioned optimizers exploits the fact that the objective functions ϕ_k are μ-strongly convex rather than just convex. We focus on denser graphs ($\delta = 4$) for which obtaining good solutions is challenging. Figure 2 illustrates Experiment 2. We notice that no matter the choice of μ, ISTA satisfies the convergence condition but fails to learn anything significant. Both FISTA and its greedy variant display similar behavior and performance as they learn significantly better solutions when μ is set high. This explains the lower performance of ProxiMAS in Gillot and Parviainen (2022) for denser graphs ($\delta = 4$) since the authors used FISTA with $\mu = 20$ in all experiments. As a rule, we observe that the larger the μ the more the convergence condition $\left\|\widetilde{W}_k - W_k\right\| \leq \left\|\widetilde{W}_k - W_{k-1}\right\|$ is satisfied. Interestingly, the behavior of the Nesterov optimizer is opposite to that of FISTA: it learns better DAGs when

μ is set smaller. Our hypothesis is that since it accounts for the μ-strong convexity of the ϕ_k objectives, it optimizes too well the MAS penalization terms $\frac{\mu}{2}\|W - W_{k-1}\|^2$, preventing progress due to new cyclic solutions $\widetilde{W}_k$ remaining too close to last acyclic solutions W_{k-1}.

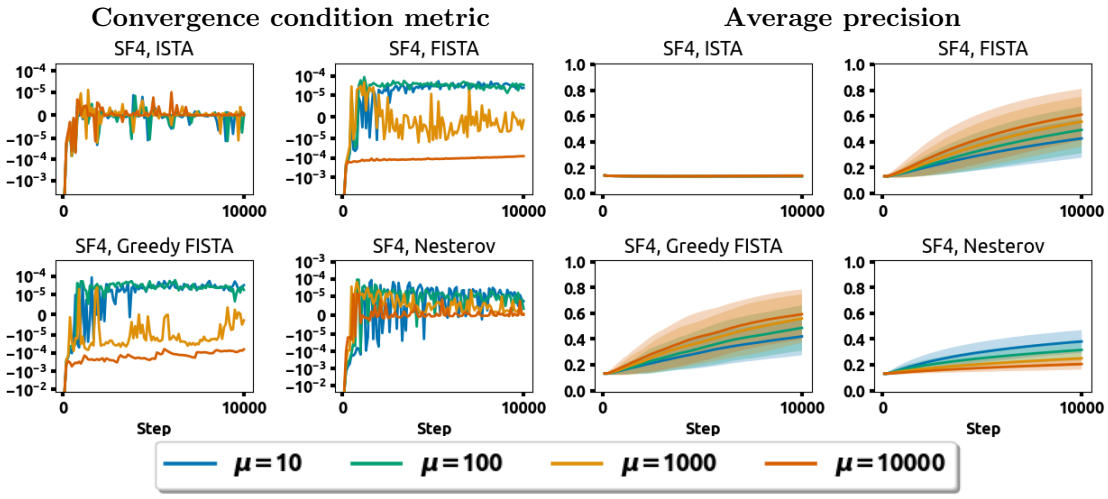

Figure 2: Experiment 2: convex optimizer and μ vary ($d=1000$, $n=10000$, NV).

4.4 Experiment 3

In the third experiment, we investigate different warmstarting strategies in order to speed-up practical convergence of FAS-based heuristics. A first form of warmstarting consists in presolving Algorithm 1 without enforcing acyclicity ($\iff \mu=0$); a second form is to first use a convex optimizer, then use a non-convex one. The hyperparameter "Cyclic %" controls the ratio of iterations dedicated to "cyclic presolving"; "Convex %" controls the ratio of iterations (excluding cyclic presolving) that use the FISTA optimizer before swapping for the adaptative optimizer Adam (Kingma and Ba, 2014) (see Table 1). For instance, assuming 20000 iterations in total, Cyclic %=50 and Convex %=20 means cyclic presolving occurs up to iteration 10000, FISTA is used up to iteration 12000, after which we use Adam. Figure 3 illustrates Experiment 3. Based on the empirical study in Gillot and Parviainen (2022), cyclic presolving is ideal when learning very sparse DAGs ($\delta \leq 2$). Experiment 3 suggests that when $\delta \geq 4$, an hybrid optimizer strategy yields superior performance boost. These boosts are more pronounced when both the number of nodes d and the number of samples n are sufficiently large. We suspect the non-linear nature of Adam makes it efficient at learning complex structures, but the non-differentiability of the ϕ_k in Algorithm 1 could explain why Adam benefits from warmstarting instead of starting from the zero matrix.

5. Discussion

We have studied theoretical convergence and demonstrated that FAS-based heuristics as presented in Gillot and Parviainen (2022) with a non-accelerated convex optimizer have

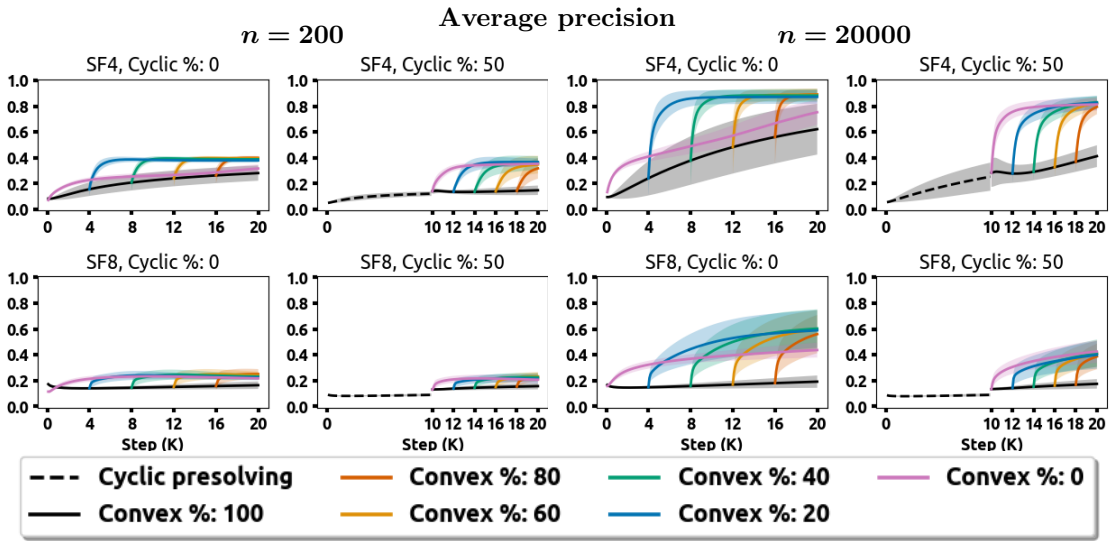

Figure 3: Experiment 3: warmstarting strategy varies ($d = 2000$, NV).

provable structural convergence (of subsequences) in a finite number of iterations, albeit a weaker form than in Park and Klabjan (2017). More specifically, additional assumptions are necessary, in the form of a) a learning rate decreasing sufficiently fast and b) a convergence condition ensuring feedback arc set costs eventually become less than the distance between past acyclic solutions and new cyclic solutions. Our empirical study provides evidence that these assumptions are mild: in practice, FAS-based heuristics are sufficiently stable in that they tend to reach a performance plateau even with constant learning rate and using an accelerated convex optimizer, thus one can decrease the learning rate only at a later stage, as a safeguard. Moreover, our study suggests that setting the hyperparameter μ sufficiently high helps satisfying the convergence condition, especially when learning denser acyclic structures. Finally, we investigated different forms of warmstarting strategies to speed-up the practical convergence of FAS-based heuristics. We uncovered an interesting effect, in that an hybrid optimizer strategy (convex optimizer followed by non-convex optimizer) consistently provides tangible acceleration when learning sufficiently dense and large DAGs.

Acknowledgments

Parts of this work have been done in the context of CEDAS (Center for Data Science, University of Bergen, UiB). The computations were performed on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway. We thank Madhumita Kundu for her valuable input.

References

A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal-recovery problems. In *Convex Optimization in Signal Processing and Communications*, page 42–88.

Cambridge University Press, 2009a.

A. Beck and M. Teboulle. A fast Iterative Shrinkage-Thresholding Algorithm with application to wavelet-based image deblurring. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 693–696, 2009b.

D. M. Chickering. Learning Bayesian Networks is NP-Complete. In *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer New York, 1996.

S. Dong and M. Sebag. From graphs to DAGs: a low-complexity model and a scalable algorithm. *CoRR*, 2022.

P. Eades, X. Lin, and W. Smyth. A fast and effective heuristic for the feedback arc set problem. In *Information Processing Letters*, volume 47, pages 319–323, 1993.

P. Gillot and P. Parviainen. Learning Large DAGs by Combining Continuous Optimization and Feedback Arc Set Heuristics. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 2022.

R. M. Karp. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.

D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2014.

J. Liang, T. Luo, and C.-B. Schönlieb. Improving "Fast Iterative Shrinkage-Thresholding Algorithm": Faster, Smarter, and Greedier. In *SIAM Journal on Scientific Computing*, volume 44, pages A1069–A1091, 2022.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer New York, 2014.

I. Ng, A. Ghassami, and K. Zhang. On the Role of Sparsity and DAG Constraints for Learning Linear DAGs. In *Advances in Neural Information Processing Systems*, 2020.

Y. W. Park and D. Klabjan. Bayesian Network Learning via Topological Order. In *Journal of Machine Learning Research*, volume 18, pages 1–32, 2017.

Y. Yu, T. Gao, N. Yin, and Q. Ji. DAGs with No Curl: An Efficient DAG Structure Learning Approach. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 12156–12166, 2021.

X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

R. Zhu, A. Pfadler, Z. Wu, Y. Han, X. Yang, F. Ye, Z. Qian, J. Zhou, and B. Cui. Efficient and Scalable Structure Learning for Bayesian Networks: Algorithms and Applications. In *IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2613–2624, 2021.