The newsletter of the Mixed Integer Programming Society

MIP Insights

Table of contents

The 2025 MIP Workshop

The **2025 Mixed Integer Programming Workshop (MIP 2025)** was held **June 2–6, 2025** at the **University of Minnesota**, returning to Minneapolis after twenty years. MIP 2025 featured

(1) a single track of 19 invited experts,
(2) a summer school preceding the main workshop,
(3) the traditional student poster session,
(4) a special session for the fourth edition of the MIP computational competition, and
(5) a set of contributed flash talks.

The scientific program was complemented by time set aside for socialization and collaboration, through tea/coffee breaks, long lunches, and organized group dinners.

MIP 2025 had 130 registered participants (including 56 students) coming from 4 continents and 13 countries. The invited speakers represented academic, national laboratory, and industry research and covered a broad range of recent advances from theoretical, computational, and applied topics; per tradition, over 50% of the presenters were early-career researchers. The summer school offered in-depth tutorials for the nearly 80 attendees on numerical optimization, column generation, and structured MINLPs by **Philipp Christophel** (SAS Institute Inc.), **Ricardo Fukasawa** (University of Waterloo), and **Andrés Gómez** (University of Southern California).

The MIP 2025 poster session, with 29 students presenting their work, was held concurrently with a reception and beautiful views from the Campus Club on the 4th floor of Coffman Memorial Union. The poster committee recognized two honorable mentions, **Dahye Han** (Georgia Tech) and **Matheus Jun Ota** (University of Waterloo), and gave the best poster award to **Matias Villagra** (Columbia University) for his work "Symmetries and Lift-and-Project Hierarchies", which was also voted the *most popular poster* by attendees. Congratulations to Dahye, Matheus, Matias, and all poster presenters, and thank you to the poster committee — chaired by Weijun Xie (Georgia Tech) — for their service!

This year, the computational competition was renamed to the **Land-Doig MIP Competition** in honor of Ailsa H. Land and Alison G. Harcourt (née Doig) for their contributions to integer programming, such as developing the branch-and-bound method. The topic of the 2025 competition was MIP Quadratic Primal Heuristics. The winning team was **Gioni Mexi, Deborah Hendrych, Sébastien Designolle, Mathieu Besançon, and Sebastian Pokutta** for their submission "A Frank-Wolfe-Based Primal Heuristic for Quadratic Mixed-Integer Optimization". Honorable mentions were given to **Yongzheng Dai** and the team of **Suri Liu and Stefan Minner**. Congratulations to all the winners! And thank you to the computational competition committee, chaired by Jan Kronqvist (KTH Royal Institute of Technology), for making the competition a success.

A new initiative at MIP 2025 was the solicitation of contributed "flash talks" by "nonstudents", adding eight technical talks to the program and creating new opportunities for participation and knowledge exchange at the workshop.

MIP 2025 was kept affordable with the support of academic, industry, and federal sponsors, including grants for student travel expenses from the National Science Foundation, Air Force Office of Scientific Research, and the Office of Naval Research. We also appreciate the local team for its smooth organization and operation of the workshop.

Visit the website and the program book to see presenter information, accepted posters, talk slides, event photos, and more.

MIP European Workshop 2025

From July 1-3 we organized the inaugural Mixed Integer Programming European Workshop at LIMOS in Clermont-Ferrand, France. With this first MIP event in Europe, we brought together 17 speakers and 67 participants. Following the formula for MIP events, we had speakers and participants from both academia and industry, covering various genders and seniority levels. We encourage everyone to view the presenter's slides on the event website. The workshop featured a poster session with 16 PhD and postdoc presenters.

The workshop brought together speakers from 8 European countries and participants from 13 countries, including many who had not attended any MIP event before due to the barriers posed by long-distance travel.

Mirroring the success of other MIP events, we brought together people interested in theory, computational work and applications of MIP. We heard much enthusiasm for repeat events in Europe in the future. We extend our thanks to our academic and industry sponsors, both locally and abroad, as well as a special thank you to the local organization (Rafael Colares and Renaud Chicoisne).

MIP South America 2025

MIP South America is just around the corner! We invite everyone to join us in Viña del Mar, Chile, for the first MIP event to be held in South America. The event will take place December 9 - 12, 2025, at Universidad Adolfo Ibáñez, and it will follow the traditional MIP single-track format of invited talks that showcase the latest advances in integer programming, discrete optimization, and related areas. The speaker lineup will feature prominent researchers from academia and industry across a wide range of fields and career stages. The program will also include a poster session. MIP South America is part of the MIP International Series.

For the full list of speakers and more information, visit the website.

We look forward to seeing you in Chile!

Program Committee: Victor Bucarey, Margarida Carvalho, Andrés Gómez, Javier Marenco, Gonzalo Muñoz, Eduardo Uchoa

Local Committee: Victor Bucarey, Rodolfo Carvajal, Gonzalo Muñoz

The 2026 MIP Workshop

The 2026 Mixed Integer Programming Workshop will take place on May 18-21, 2026, at the University of Connecticut, Stamford campus. This will be the twenty-third edition of the series. MIP 2026 will carry on the traditions that have made this workshop a central gathering place for our community, and we invite you to join us for:

- A single track of 20 invited talks from experts across a wide range of topics in the theoretical, computational, and applied aspects of mixed integer programming and discrete optimization.
- A poster session, including a best-poster competition among selected student finalists. More information can be found here.
- The 2026 Land-Doig MIP Competition on GPU-Accelerated Primal Heuristics for MIP. More information can be found here.

For updates and more information, please visit the website.

We look forward to seeing you in Stamford, Connecticut!

Program committee: Beste Basciftci, Yatharth Dubey, Cheng Guo, Sebastian Perez-Salazar, Matthias Walter

Local committee: David Bergman, Carlos Cardonha, Robert Day, Nicholas Lownes, Laurent Michel, Matthew Stuber, Bin Zou

Competition Committee: Beste Basciftci, Akif Çördük, Gerald Gamrath, Christopher Hojny, Jan Kronqvist, Haihao Lu, Christian Tjandraatmadja

Call for MIP locations

If you are interested in hosting MIP 2027, please send an email to sperez@rice.edu. We kindly ask you state your interest by February 1, 2026, at which time the program committee will start to evaluate potential locations. The committee will continue to evaluate potential locations until an appropriate one is chosen.

Call for MIP International locations

Following the success of the MIP International events, we aim to support additional workshops in this series. We therefore invite you to express your interest in hosting a future workshop by sending an email to mail@mixedinteger.org. Interested applicants may be asked for a short proposal.

Discrete Optimization Talks (DOTs)

The Mixed Integer Programming Society supports Discrete Optimization Talks (DOTs), a virtual seminar series on all aspects of integer and combinatorial optimization. This semester, DOTs will be take place on the first Friday of the month at 12:00pm Eastern Time over Zoom. Visit our website to find information on the Fall 2025 season of DOTs and view recordings of previous talks. To receive the link to participate, join the mailing list and add "lists@mixedinteger.org" to your approved addresses. If you are interested in giving a DOT, let us know. We look forward to seeing you!

Call for ISMP Locations

The Conference Committee of the Mathematical Optimization Society (the parent organization for the Mixed Integer Programming Society) issues a call for pre-proposals to organize and host ISMP 2030, the 27th triennial International Symposium on Mathematical Programming. According to tradition, we are particularly seeking a host outside of Europe for ISMP 2030.

ISMP is the flagship event of our society, regularly gathering over a thousand scientists from around the world. The conference will be held in or around the month of August, 2030. Hosting ISMP provides a vital service to the mathematical optimization community and often has a lasting effect on the visibility of the hosting institution. It also presents a significant challenge. This call for pre-proposals is addressed at local groups willing to take up that challenge.

Preliminary bids will be examined by the Conference Committee, which will then issue invitations for detailed bids. The final decision will be made and announced during ISMP 2027 in Amsterdam.

Member of the MOS Conference Committee are

Jeff Linderoth, Chair <linderoth@wisc.edu>

Karen Aardal, <K.I.Aardal@tudelft.nl>

Aleksandr Kazachkov, <akazachkov@ufl.edu>

Defeng Sun, <defeng.sun@polyu.edu.hk>

Wolfram Wiesemann, <ww@imperial.ac.uk>

Katya Scheinberg, ex-officio <katyascheinberg@gmail.com>

Preliminary bids should be brief and contain information pertaining to the

1) Location,
2) Facilities,
3) Logistics: accommodation and transportation, and
4) Likely local organizers.

Further information can be obtained from any member of the advisory committee. Please address your preliminary bids until November 15, 2025 to Jeff Linderoth <linderoth@wisc.edu>.

Discrete optimization in Statistics

By Sanjeeb Dash IBM Research, Yorktown Heights, NY 10598

Many recent papers propose mixed-integer programming (MIP) formulations of important problems in statistics and machine learning such as the problem of finding 'interpretable' classifiers from data. The connection with MIP arises from the assumption that interpretability and generalizability are linked to sparsity. Some recently studied interpretable classifiers are sparse decision trees (the number of nodes in the tree is small) [4, 17, 16, 15] and sparse boolean decision rule sets (equivalently, a boolean formula in disjunctive normal form where the total number of literals in the formula is small) [27, 12].

Sparse linear regression [22, 5] is another important area of study as linear regression is a fundamental tool in statistics. The goal of the linear regression problem is to find the 'best' parameters in a linear model $y = \sum_{i=1}^n \beta_j X_j + \beta_0$, where $X_1, \ldots, X_n$ are n independent variables, y is a dependent variable, and $\beta_0, \ldots, \beta_n$ are unknown model parameters. For ease of exposition, we will assume $\beta_0 = 0$ and only discuss linear models of the form

$$y = \sum_{i=1}^{n} \beta_j X_j. \tag{1}$$

Let $\mathbf{y} \in \mathbb{R}^m$ and let $\mathbf{X}$ be an $m \times n$ matrix of real numbers (also called a *design matrix*). Assume the rows of $\mathbf{y}$ are the values of y corresponding to the values of $X_1, \ldots, X_n$ given by rows of $\mathbf{X}$. The least-squares estimation problem is

$$\min_{\beta \in \mathbb{R}^n} \|\mathbf{y} - \mathbf{X}\beta\|_2. \tag{2}$$

In many applications, the least-squares estimation problem is solved to find the 'best' β in the linear model in (1). When the columns of $\mathbf{X}$ are linearly independent, there is a *unique solution* to the least-squares problem. There are other metrics besides $\|\mathbf{y} - \mathbf{X}\beta\|_2$ that are used to measure the quality of model fit for given β. For example, when the X_is are assumed to be independent Gaussian variables, ϵ is a Gaussian noise variable with zero mean, and $y = \sum_{j=1}^n \beta_j X_i + \epsilon$, the *likelihood* function $\mathcal{L}(\beta)$ – which measures the likelihood of the model parameters β given the data – may be used. In this case, the β that maximizes the likelihood function is the desired solution. Under some commonly used statistical assumptions, the maximum likelihood solution and the least-squares solution are the same: $\arg\min_\beta \|\mathbf{y} - \mathbf{X}\beta\|_2 = \arg\max_\beta \mathcal{L}(\beta) = \arg\max_\beta \log(\mathcal{L}(\beta))$.

In one variant of sparse linear regression, the goal is to get a solution of the linear regression problem with the restriction that $\|\beta\|_0 \leq t$ for some positive integer $t < n$ where $\|.\|_0$ is a function that returns the number of nonzero components in its argument. One way to attain this goal is to solve a sparse least-squares problem of the form

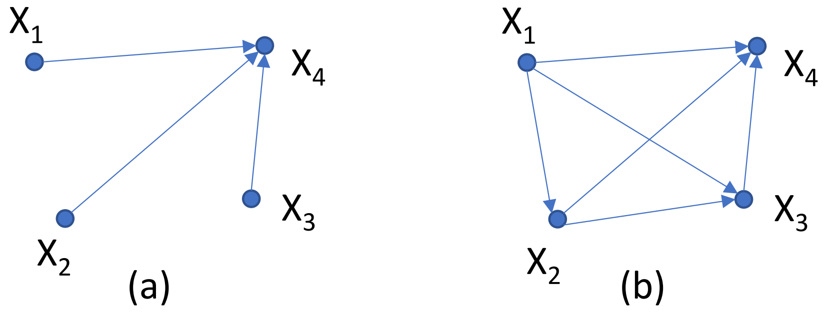
$$\min_{\beta \in \mathbb{R}^n} \|\mathbf{y} - \mathbf{X}\beta\|_2 \text{ subject to } \|\beta\|_0 \leq t. \tag{3}$$

See [14, 2, 5] for mixed-integer and semidefinite optimization techniques for solving (3). Adding the constraint $\|\beta\|_0 \leq t$ transforms the problem from being easily solvable to an NP-hard problem [23]. Other complications arise when the columns of $\mathbf{X}$ are linearly dependent. Some variants of (3) are $\min_{\beta \in \mathbb{R}^n} \|y - \mathbf{X}\beta\|_2 + \lambda\|\beta\|_0$ or $\max_\beta \log(\mathcal{L}(\beta)) - \lambda\|\beta\|_0$ for some λ.

We next discuss a generalization of sparse linear regression that is not well-studied from the perspective of designing MIP-based solution methods. Linear Structural Equation Models (SEM) – which are very popular in the social sciences (e.g., psychology) [26] – are a way of representing joint probability distributions over a set of Gaussian random variables by *directed mixed-graphs* (DMG), i.e., graphs with directed edges (indicated by (i, j) where i, j are nodes or by an arrow $i \to j$) and bidirected edges (indicated by $\{i, j\}$ or by $i \leftrightarrow j$). Each node represents a model variable; some nodes may represent *latent* or unobserved variables. High-quality software such as LISREL [20, 21] and AMOS [1] are available to learn these types of 'graphical models' from data.

We represent a DMG as $G = (V, E_d, E_b)$ with node set V, directed edge set $E_d \subseteq \{(i, j) : i, j \in V, i \neq j\}$ and bidirected edge set $E_b \subseteq \{\{i, j\} : i, j \in V, i \neq j\}$. We call node i an ancestor of node j in G if there is a directed path from i to j in G or $i = j$. We call node i a parent of node j in G if there is a directed edge from i to j in G. We denote the set of ancestors and the set of parents of node i in G by $\text{an}_G(i)$ and $\text{pa}_G(i)$, respectively. Similarly, given a set $D \subset V$ of nodes in G, we denote the union of parents of nodes in D by $\text{pa}_G(D)$. If G is clear from the context, we drop the subscript G. An acyclic DMG (or ADMG) contains no directed cycle ($i \to k \to \ldots \to j \to i$). An ADMG without bidirected edges is just a directed acyclic graph (DAG). An ADMG is called an *AADMG* (*maximal ancestral ADMG*) if it does not contain an *almost directed cycle* $i \to k \to \ldots \to j \leftrightarrow i$ (i.e., $\{i, j\} \in E_b$ is a bidirected edge and $i \in \text{an}_G(j)$). We next discuss SEMS representable by directed graphs.

In an SEM, a directed edge from X_i to X_j indicates the presence of an equation where X_j is a linear combination of variables including X_i. For example, the graph (a) below indicates a linear model of the form $X_4 = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon_4$, along with the equations $X_i = \epsilon_i$ $(i = 1, \ldots, 3)$, where the ϵ_is are independent random variables that follow a normal distribution. The above linear relationship implies that $X_1, \ldots, X_4$ follow a multivariate normal distribution. In this case, if we draw m samples from this multivariate normal distribution, create the vector $\mathbf{y}$ from sample values for X_4, and create the design matrix $\mathbf{X}$ from sample values of X_1, X_2, X_3, then $\mathbf{X}$ will have linearly independent columns with high probability and the least squares problem in (2) or (3) will have a unique solution for $\beta_1, \beta_2, \beta_3$.



The first way in which SEMs generalize linear regression is that they assume multiple dependent variables, not just one. Let $G = (V, E_d, \emptyset)$ be the directed graph (b) above. Instead of a vector β of parameters, assume there is a 4×4 matrix of parameters B and that the graph indicates the presence of the equations:

$$\begin{aligned} X_1 &= \epsilon_1 \\ X_2 &= B_{21}X_1 + \epsilon_2 \\ X_3 &= B_{31}X_1 + B_{32}X_2 + \epsilon_3 \\ X_4 &= B_{41}X_1 + B_{42}X_2 + B_{43}X_3 + \epsilon_4. \end{aligned} \tag{4}$$

As before, assume $\epsilon_1, \ldots, \epsilon_4$ are independent Gaussian random variables with mean 0; their covariance matrix Ω is a diagonal matrix. Then $X_1, \ldots, X_4$ follow a multivariate normal distribution. Note that each variable depends linearly on lower numbered variables. B_{ij} is nonzero only if there is an edge from j to i. Then B is lower-triangular as G is a directed acyclic graph (DAG) and directed edges go from lower numbered variables to higher numbered ones. We could try estimating the nonzero coefficients in B by solving three linear regression problems, one each for X_2, X_3, and X_4, and then estimate Ω. In a sense, the problem decomposes into four separate estimation problems, one per variable (for X_1, we only need to estimate its variance). Just as correlations between variables make linear regression problems harder to solve, the estimation problem for B becomes more complicated when there is a cycle. In this case, that there are many possible choices for B which lead to equivalent multivariate distributions; B is then not (uniquely) *identifiable*.

More generally, an SEM representable by a DAG is defined by random variables $X_1, \ldots, X_n$ and an $n \times n$ matrix B along with the following linear models which connect (possibly after a permutation of variables) each variable X_i to previous variables $X_1, \ldots, X_{i-1}$:

$$X_i = \sum_{j<i} b_{ij} X_i + \epsilon_i \tag{5}$$

where $\epsilon_1, \ldots, \epsilon_n$ are independent Gaussian random variables with a diagonal covariance matrix Ω. B is clearly lower triangular. If we defined a directed graph G with a directed edge from j to i if and only $B_{ij} \neq 0$, then G is a DAG. Conversely, if we start out with a DAG and define nonzero B_{ij} coefficients if and only if there is an edge from j to i, then we can permute the indices (via a topological sort) such that the same permutation makes B a lower triangular matrix. The number of parameters in the model equals $|E_d| + 2n$; there is one β_{ij} for every directed edge ji, and a mean and variance parameter per ϵ_i. We assume all means are zero; then the number of unknown model parameters is $|E_d| + n$. Then $X = (X_1, \ldots, X_n)$ follows a multivariate normal distribution $N(0, \Sigma)$ where the covariance matrix Σ is given by

$$\Sigma = (I - B)^{-T} \Omega (I - B)^{-1}. \tag{6}$$

We say that the distribution $N(0, \Sigma)$ (or matrix Σ) is consistent with G in the sense that it follows from (6) by choosing appropriate B and Ω where the nonzero pattern in B matches the pattern of edges in G. When G is a DAG, the relationship between (B, Ω) – where Ω is a diagonal matrix with positive diagonal entries and $B_{ij} \neq 0$ only if $ij \in E_d$ – and the resulting Σ (in (6) is a bijection and B, Ω are said to be *identifiable* given a Σ consistent with G.

An important problem in statistics is to start out with m samples drawn from a multivariate distribution defined by an **unknown DAG-representable SEM** and then try to learn the associated DAG G and parameter matrices B and Ω. We call this the *DAG learning problem* for SEMs. A multivariate normal distribution $N(0, \Sigma)$ is fully defined by its covariance matrix Σ. Let $\hat{\Sigma}$ be the sample covariance matrix. In essence, the DAG learning problem attempts to learn matrices B and Ω where Ω is diagonal with all diagonal entries positive, B is lower-triangular, possibly after permuting its rows and columns, and

$$\hat{\Sigma} \approx (I - B)^{-T} \Omega (I - B)^{-1} \text{ or } \hat{\Sigma} \approx \Sigma, \tag{7}$$

Though it would seem natural to use $\|\hat{\Sigma} - \Sigma\|$ as a measure of quality of fit, where $\| \cdot \|$ is some matrix norm, there are many different scoring functions which measure how well the parameters B, Ω (or the resulting distribution $N(0, \Sigma)$) fit the data. As the sample covariance matrix $\hat{\Sigma}$ may differ from Σ, it may not be possible to recover the ground-truth Σ (in the sense of equation (6)).

Bayesian networks form another class of statistical models represented by DAGs. Each DAG G encodes a factorization of the joint probability distribution $p(X_1, \ldots, X_n)$ in terms of conditional probability distributions as follows:

$$p(X_1, \ldots, X_n) = \Pi_{i=1}^n p(X_i \,|\, pa_G(X_i)).$$

Just as sparse linear regression is a natural extension of linear regression, it is common to search for sparse DAGs that fit the data well (for both SEMs and Bayesian networks). The most common type of sparsity considered is a uniform bound on the indegree of each DAG node. Note that if only a single fixed node in the graph is assumed to have positive indegree (as in graph (a)), then the sparse DAG learning problem for SEMS is essentially a sparse linear regression problem! Therefore the general sparse DAG learning problem for SEMs can be viewed as a collection of linked sparse linear regression problems.

The BIC score (Bayesian Information Criterion) is an important score function used to measure how well an SEM (with graph representation G) or a Bayesian network fits given data. It penalizes the number of edges in the solution DAG G; for SEMs,

$$\text{BIC}(G) = \log(\max_{B,\Omega} \mathcal{L}(B, \Omega)) - t \log(m)/2,$$

where $\mathcal{L}(B, \Omega)$ measures the likelihood of the parameters B, Ω (defined over G) given the data, t is the number of parameters (= $|E| + n$), and m is the number of samples. One usually attempts to find a bounded indegree DAG which maximizes the BIC score. RMSEA (root mean square error of approximation) is another popular [25] measure of quality of model fit to data. The AMOS software [18] reports over a dozen model fit metrics.

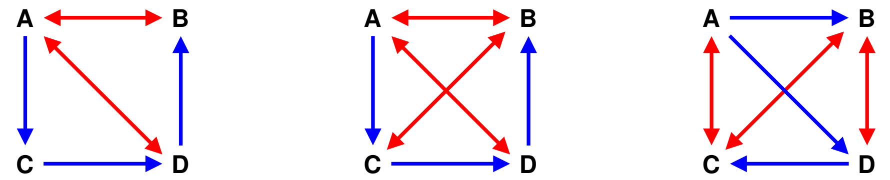
For a DAG G with n nodes, let $G^i = (V^i, E^i)$ for $i = 1, \ldots, n$, where $V^i = \{i\} \cup pa_G(i)$ and $E^i = \{ji : j \in pa_G(i)\}$. In other words, G^i is a subgraph of G formed by taking node i and the edges joining the parents of i to i; we call such a structure an *in-star*. For example, graph (a) depicts an-star rooted at X_4, while graph (b) decomposes into 4 in-stars, one per node. An important property of the BIC score for a DAG G is that $\text{BIC}(G) = \sum_{i=1}^n \text{BIC}(G^i)$. This decomposition is the basis for an MIP formulation and branch-and-bound method to find a bounded indegree DAG with maximum BIC score; the ideas behind this formulation were developed in [19, 7, 10, 11, 3]. A detailed polyhedral study can be found in [11] and a branch-and-bound code GOBNILP [3, 8] implements these ideas. GOBNILP takes as input a list of in-stars (typically set to all in-stars with degree at most t for some t, say 3 or 4) that can be defined on n nodes (there are $n\binom{n-1}{t}$ such in-stars), computes the BIC score of each input in-star, defines a binary variable per in-star, and then creates an MIP formulation that chooses one in-star per node such that the union of chosen edges forms a DAG and the sum of BIC scores is as large as possible. With this approach, one can solve DAG learning problems for problems with a hundred to a few hundred nodes if one assumes the indegree is bounded by 2 and around a hundred nodes if the bound is 3. Higher bounds are hard to handle in practice. Column generation has recently been used to deal with the large number of in-star variables [9, 28]; however the pricing problem is nontrivial.

We get even more complicated problems if we assume that the random variables $\epsilon_1, \ldots, \epsilon_n$ in an SEM are correlated (a correlation between ϵ_i and ϵ_j is often assumed to be because of an unobserved/latent variable). In other words, assume $\epsilon_1, \ldots, \epsilon_n$ are drawn from $N(0, \Omega)$, where Ω has nonzero off-diagonal elements. In such a situation, the SEM graph has both directed and bidirected edges. We assume that the nonzero entry pattern in B and Ω is modeled by $G = (V, E_d, E_b)$ that satisfies

$$B_{ij} = 0 \text{ if } i = j \text{ or } j \to i \text{ is not in } E_d, \tag{8}$$

$$\Omega_{ij} = 0 \text{ if } i \leftrightarrow j \text{ is not in } E_b. \tag{9}$$

Just as the DAG condition yields a unique decomposition of Σ in terms of B and Ω in (6), certain restrictions in the form of forbidden subgraphs are imposed on DMGs to get this unique decomposition property. A *bow-free* graph is an ADMG that does not have a pair of nodes a, b such that $a \rightarrow b$ and $a \leftrightarrow b$ (such a structure is called a bow). The definition of an *arid* graph is more involved. Given a set of nodes $S \subseteq V$, we say a rooted arborescence on S is a subgraph T of $G[S]$ (the subgraph of G induced on S) consisting of directed edges that define a rooted, directed tree on S: there is one node r in T with outdegree 0 but positive indegree, all other nodes in T have outdegree 1, and T is acyclic and connected. We say that an ADMG is arid if there is no subset of nodes S in G such that the induced bidirected subgraph of G on S forms a connected component, and the induced directed subgraph on S contains a rooted arborescence spanning S. Note that an arid graph does not contain a bow, and therefore all arid graphs are bow-free. Further, any AADMG is arid. See the figure below. Graphs (i) and (iii) are bow-free and arid but not AADMGs; (ii) is bow-free but not arid and not an AADMG.



(i) (ii) (iii)

For an arid graph, a consistent covariance matrix Σ always has a unique decomposition in terms of B and Ω matrices in equation (6), whereas this is almost always true for bow-free graphs. However, a Gaussian distribution can be consistent with multiple bow-free graphs and there is no complete theory [24] identifying this class of graphs. Given a finite set of input data samples from $N(0, \Sigma)$, an MIP-based solution approach for the problem of finding a bow-free (arid) ADMG G such that there exists associated (B, Ω) satisfying equations (8),(9) and BIC(G) is as large as possible was given in [13]. A similar score-maximization problem for the class of AADMGs was solved via MIP techniques in [6].

How can one find an arid graph, bow-free graph, or AADMG with maximum BIC score? In the DAG learning problem, the BIC score decomposed into a sum of BIC scores of simpler subgraphs (namely in-stars). The BIC score of an ADMG can be decomposed into the sum of BIC scores of *c-components*. One can then consider 'bounded' size c-components, calculate the BIC score per c-component, and then find the set of c-components that can be glued together to form an arid graph, bow-free graph, or an AADMG with maximum possible BIC score among all that can be formed from bounded size c-components. We define c-components next.

Given a DMG G, the *district* for node i is the set of nodes in the component containing node i in the subgraph of G induced by all bidirected edges. In a DAG, each district consists of a single node, and each node forms a single-node district. Given a district $\mathcal{D}$ of G, we define a DMG $G_{\mathcal{D}}$ with node set $\mathcal{D} \cup \text{pa}_G(\mathcal{D})$. The bidirected edges in $G_{\mathcal{D}}$ are the bidirected edges in G that connect pairs of nodes in $\mathcal{D}$. The directed edges in $G_{\mathcal{D}}$ are the directed edges in G that point to nodes in $\mathcal{D}$. We call $G_{\mathcal{D}}$ a *c-component*. Letting G be the graph (i) in the figure above, we see that the nodes and edges can be partitioned into two c-components, one consisting of the district $\{C\}$ and the edge AC, and the other consisting of the district $\{A, B, D\}$ and all edges in G other than AC. The BIC score of G is the sum of BIC scores of these two c-components. In the solution approach in [6] and [13], a bound on the number of nodes in a district and a bound on the number of parent nodes per district node is taken as input. Then all c-components satisfying these bounds are enumerated and the BIC score per c-component is calculated (a nontrivial optimization task). Then an MIP with a binary variable per c-component and constraints that enforce the bow-free condition (or arid condition or AADMG condition) is solved. This approach can solve learning problems with just about 10-25 nodes if district sizes are at most 2 and parent set sizes are at most 3; the number of bounded size c-components grows very rapidly with the number of nodes. Even some published heuristics for bow-free graph learning take a lot of time – many hours to learn 20 node graphs.

There is a very rich set of problems that are not yet solved in this area of learning graphical models. We have discussed a few classes of graphs that are relevant for linear SEMs defined on Gaussian random variables. Different types of structures are relevant for other distributions. Learning graphs with optimal scores on more than 20-30 nodes is a very difficult task, especially if we do not impose any bounds on the c-component sizes. Some scoring functions such as BIC scores decompose into the sum of BIC scores of c-components, yet some scores are not decomposable. Tractable optimization formulations for many of the above problems remain to be found.

[1] J.L. Arbuckle. Amos: Analysis of moment structures. *The American Statistician*, pages 66–67, 1989.

[2] A. Atamtürk and A. Gómez. Rank-one convexification for sparse regression. *Journal of Machine Learning Research*, 26:1–50, 2025.

[3] M. Bartlett and J. Cussens. Integer linear programming for the bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017.

[4] D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7): 1039–1082, 2017.

[5] D. Bertsimas, A. King, and R. Mazumder. Best subset selection via a modern optimization lens. *Annals of Statistics*, 44 (2):813–852, 2016.

[6] R. Chen, S. Dash, and T. Gao. Integer programming for causal structure learning in the presence of latent variables. In *International Conference on Machine Learning*, 2021.

[7] J. Cussens. Bayesian network learning with cutting planes. In *Conference on Uncertainty in Artificial Intelligence*, page 153–160, 2011.

[8] J. Cussens. Gobnilp: Learning bayesian network structure with integer programming. In *International Conference on Probabilistic Graphical Models*, pages 605—608, 2020.

[9] J. Cussens. Branch-price-and-cut for causal discovery. In *Conference on Causal Learning and Reasoning*, page 642–661, 2023.

[10] J. Cussens, D. Haws, and M. Studenỳ. Polyhedral aspects of score equivalence in Bayesian network structure learning. *Mathematical Programming*, pages 1–40, 2016.

[11] J. Cussens, M. Járvisalo, J. H. Korhonen, and M. Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *Journal of Artificial Intelligence Research*, 58:185–229, 2017.

[12] S. Dash, O. Günlük, and D. Wei. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*, volume 31, 2019.

[13] S. Dash, J. Goncalves, and T. Gao. Integer programming based methods and heuristics for causal graph learning. In *International Conference on Artificial Intelligence and Statistics*, 2025.

[14] A. Del Pia and D. Zhou. A semidefinite programming relaxation for the sparse integer least squares problem. *Informs Journal on Optimization*, 2025. doi: 10.1287/ijoo.2023.0003.

[15] E. Demirovic, A. Lukina, E. Hebrard, J. Chan, J. Bailey, C. Leckie, K. Ramamohanarao, and P. J. Stuckey. Murtree: Optimal decision trees via dynamic programming and search. *Journal of Machine Learning Research*, 23:1–47, 2022.

[16] O. Günlük, J. Kalagnanam, M. Li, M. Menickelly, and K. Scheinberg. Optimal decision trees for categorical data via integer programming. *Journal of Global Optimization*, 81: 233–260, 2021.

[17] X. Hu, C. Rudin, and M. Seltzer. Optimal sparse decision trees. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019.

[18] IBM. IBM SPSS Amos, version 29, 2022. URL https://www.ibm.com/products/structural-equation-modeling-sem

[19] T. Jaakkola, D. Sontag, A. Globerson, and M. Meila. Learning bayesian network structure using lp relaxations. In *International Conference on Artificial Intelligence and Statistics*, pages 358–365, 2010.

[20] K.G. Jöreskog. A general method for estimating a linear structural equation system. In *Structural Equation Models in the Social Sciences, A.S. Goldberger and O. D. Duncan (eds.)*, pages 83–112. Academic Press, New York, 1973.

[21] K.G. Jöreskog and D. Sörbom. *LISREL: A General Computer Program for Estimation of a Linear Structural Equation System by Maximum Likelihood Methods*. National Educational Resources, Chicago, 1978.

[22] A. Miller. *Subset Selection in Regression*. Chapman and Hall/CRC, 2002.

[23] B. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[24] C. Nowzohour, M.H. Maathuis, R.J. Evans, and P. Bühlmann. Distributional equivalence and structure learning for bow-free acyclic path diagrams. *Electronic Journal of Statistics*, 11(2):5342–5374, 2017.

[25] V. Savalei. Improving fit indices in structural equation modeling with categorical data. *Multivariate Behavioral Research*, 56:390–407, 2021.

[26] P. Tarka. An overview of structural equation modeling: its beginnings, historical development, usefulness and controversies in the social sciences. *Qual Quant.*, 52:313–354, 2018.

[27] T. Wang, C. Rudin, F. Doshi-Velez, Y. Liu, E. Klampfl, and P. MacNeille. A bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, pages 1–37, 2017.

[28] Y. Yang and R. Chen. Inexact column generation for bayesian network structure learning via difference-of-submodular optimization. In *Advances in Neural Information Processing Systems*, 2025.

In 2022, the Mixed Integer Programming Society (MIPS) was established as a technical section of the Mathematical Optimization Society. In the MIP Insights *newsletter, we announce important news for the community, promote and summarize events supported by the society, and host expository presentations. Sanjeeb Dash, Silvia Di Gregorio, Sophie Huiberts, Aleksandr M. Kazachkov, Gonzalo Muñoz, Joseph Paat, and Sebastian Perez-Salazar contributed to this issue.*