Electronic Companion to Network Relaxations for Combinatorial Bilevel Optimization under Linear Interactions

Leonardo Lozano David Bergman Andre A. Cire

A Proofs

Proof of Proposition 1. Let $\mathbf{p}_1, \ldots, \mathbf{p}_K$ be the paths in $\mathcal{N}$ starting from the root r and ending at a terminal node, where $K \geq 1$. For every path $k \in \{1, \ldots, K\}$, we will construct a point $(\mathbf{x}^{(k)}, z^{(k)}, \boldsymbol{\omega}^{(k)}) \in \mathscr{P}[\mathcal{N}]$, and show that the collection of such points are the extreme points of $\mathscr{P}[\mathcal{N}]$. The result then follows by Definition 1, since there is a one-to-one mapping between root-terminal paths of $\mathcal{N}$ and $\mathscr{G}$.

Consider an arbitrary path $\mathbf{p}_k$ and its edges $(u_1^{(k)}, u_2^{(k)}, x_1^{(k)}), \ldots, (u_{n_l}^{(k)}, u_{n_l+1}^{(k)}, x_{n_l}^{(k)})$. Define $\mathbf{x}^{(k)}$ by collecting the corresponding edge labels, that is,

$$\mathbf{x}^{(k)} = \left(x_1^{(k)}, \ldots, x_{n_l}^{(k)}\right), \tag{1}$$

which is a binary vector. Next, define the flow vector $\boldsymbol{\omega}^{(k)} \in \{0,1\}^{|\mathscr{E}|}$ as

$$\omega_e^{(k)} = \begin{cases} 1, \text{ if path } \mathbf{p}_k \text{ includes edge } e, \\ 0, \text{ otherwise,} \end{cases} \qquad \forall e \in \mathscr{E}.$$

Note that, since the path includes exactly one edge per layer, it follows by the construction above that (7c) and (7d) are satisfied, i.e.,

$$x_j^{(k)} = \sum_{e=(u,u',1):\, u \in \mathscr{U}_j} \omega_e^{(k)} = 1 - \sum_{e=(u,u',0):\, u \in \mathscr{U}_j} \omega_e^{(k)}, \qquad \forall j \in [n_l].$$

Analogously, inequalities (7a), (7b), and (7f) hold by construction since $\mathbf{p}_k$ is a path in $\mathcal{N}$. Finally, we can set

$$z^{(k)} = \sum_{u \in \mathscr{U}_{n_l+1}} \sum_{e \in \mathscr{E}^-(u)} \nu_u \, \omega_e^{(k)}$$

to satisfy (7e), where we note that $z^{(k)} = \phi(\mathbf{x}^{(k)})$ by Definition 1.

Since each path of $\mathcal{N}$ differs by at least one edge, two paths cannot be mapped to the same vector $(\mathbf{x}^{(k)}, z^{(k)}, \boldsymbol{\omega}^{(k)})$ by construction of $\boldsymbol{\omega}^{(k)}$. Similarly, the collection of vectors $\mathbf{x}^{(k)}$ will be distinct, which implies that one cannot be written as a convex combination of others since they are binary vectors. That is, the collection of such points are extreme in $\mathscr{P}[\mathcal{N}]$ and have a one-to-one correspondence with paths in $\mathcal{N}$.

It remains to show that $\mathscr{P}[\mathcal{N}]$ does not contain any other extreme point. Consider any (possibly fractional) point $(\mathbf{x}', z', \boldsymbol{\omega}') \in \mathscr{P}[\mathcal{N}]$. Since $\boldsymbol{\omega}'$ satisfies the network-flow constraints (7a), (7b), and (7f), there exists a decomposition of $\boldsymbol{\omega}'$ such that, for any edge $e = (u, u', x) \in \mathscr{E}$,

$$\begin{aligned} \omega'_e &= \sum_{k \in [K]:\, \mathbf{p}_k \text{ includes } e} \alpha_k \\ &= \sum_{k=1}^{K} \alpha_k\, \omega_e^{(k)}, \end{aligned}$$

for some $\alpha_k \geq 0$ for all k (Theorem 3.5, Ahuja et al. 1993). Moreover, from (7a),

$$1 = \sum_{e \in \mathscr{E}^+(r)} \omega_e = \sum_{e \in \mathscr{E}^+(r)} \sum_{k=1}^{K} \alpha_k\, \omega_e^{(k)} = \sum_{k=1}^{K} \alpha_k,$$

where the last equality follows because all the arcs in the first layer correspond to the arcs leaving the root node r. Therefore, $\boldsymbol{\omega}'$ is a convex combination of the vectors $\boldsymbol{\omega}^{(k)}$ defined by the paths of $\mathcal{N}$. Similarly, for all $j \in [n_l]$,

$$\begin{aligned} x'_j &= \sum_{e=(u,u',1):\, u \in \mathscr{U}_j} \omega'_e \\ &= \sum_{e=(u,u',1):\, u \in \mathscr{U}_j} \left(\sum_{k=1}^{K} \alpha_k\, \omega_e^{(k)} \right) \\ &= \sum_{k=1}^{K} \alpha_k \sum_{e=(u,u',1):\, u \in \mathscr{U}_j} \omega_e^{(k)} \\ &= \sum_{k=1}^{K} \alpha_k x_j^{(k)}, \end{aligned}$$

and analogously for (7d) and for z' in (7e). Thus, $(\mathbf{x}', z', \boldsymbol{\omega}')$ is a convex combination of the points $(\mathbf{x}^{(k)}, z^{(k)}, \boldsymbol{\omega}^{(k)})$ defined by the paths of $\mathcal{N}$. Since this point is arbitrary, the collection of such points define the extreme points of $\mathscr{P}[\mathcal{N}]$. ■ □

Proof of Proposition 3. The symmetry result follows by backward induction on the layer j. The basis case $j = n_l+1$ holds because nodes $u, v \in \mathscr{U}_{n_l+1}$ are symmetric if and only if $\nu_u = \nu_v$, captured by (R1). For some $j < n_l + 1$, suppose no two nodes in layers $\mathscr{U}_{j+1}, \ldots, \mathscr{U}_{n_l+1}$ are symmetric after applying Algorithm 1, but nodes $u, v \in \mathscr{U}_j$ at layer j are symmetric and violate condition (R2).

There exists some $x \in \{0, 1\}$ such that (u, u', x), $(v, v', x) \in \mathscr{E}$ for nodes $u', v' \in \mathscr{U}_{j+1}$, $u' \neq v'$. However, by Definition 2, u' and v' are equivalent. Otherwise, there must exist some path starting at u to a terminal node that either has distinct labels, or terminal values with all paths starting at v and also ending at a terminal node. This contradicts the induction hypothesis. ■ □

Proof of Proposition 4. We show (i) and (ii) by an induction in the layer index j that iteratively constructs the desired mapping. For the base case $j = 1$, it suffices to set $\mathcal{T}_1 = \{\mathbf{0}\}$. For any $j \in [n_l]$, assume such a mapping exists and condition (i) is satisfied by the nodes of $\mathcal{N}$ for layers $1, \dots, j$. Then, for each node $u' \in \mathscr{U}_{j+1}$ in the $(j+1)$-th layer, define

$$\mathcal{T}_{j+1,u'} = \left\{\chi_j + \mathbf{a}_j x \colon \forall (u, u', x) \in \mathscr{E},\ \forall \chi_j \in \mathcal{T}_{j,u}\right\},$$

which satisfies (i) by construction. Moreover, by induction hypothesis, for any generated $\chi_j + \mathbf{a}_j x \in \mathcal{T}_{j+1,u'}$ there exists $x' \in \{0,1\}^{j-1}$ such that

$$\chi_j + \mathbf{a}_j x = \sum_{i=1}^{j-1} \mathbf{a}_i x'_i + \mathbf{a}_j x = \sum_{i=1}^{j} \mathbf{a}_i x,$$

which implies that $\chi_j + \mathbf{a}_j x \in \mathcal{T}_{j+1}$, and hence $\mathcal{T}_{j+1,u'} \subseteq \mathcal{T}_{j+1}$.

It remains to show that this mapping satisfies condition (ii) for every terminal node $u' \in \mathscr{U}_{n_l+1}$. Consider any two $\chi, \chi' \in \mathcal{T}_{n_l+1,u'}$. By construction, there exists $\mathbf{x}, \mathbf{x}' \in \{0,1\}^{n_l}$ such that $\chi = \sum_{i=1}^{j} \mathbf{a}_i x_i = \mathbf{A}\mathbf{x}$ and $\chi' = \sum_{i=1}^{j} \mathbf{a}_i x'_i = \mathbf{A}\mathbf{x}'$. Thus,

$$\overline{\phi}(\chi) = \overline{\phi}(\mathbf{A}\mathbf{x}) = \phi(\mathbf{x}) = \nu_{u'} = \phi(\mathbf{x}') = \overline{\phi}(\mathbf{A}\mathbf{x}') = \overline{\phi}(\chi'),$$

where the second equality follows the definition of $\overline{\phi}$, the third equality follows from Definition 1-(C), and the remaining equalities are symmetric.

We next show (iii). Let $u, u' \in \mathscr{U}_j$ be two distinct nodes in the j-th layer such that there exists $\chi \in \mathcal{T}_{j,u} \cap \mathcal{T}_{j,u'}$. If $j = n_l + 1$, then u, u' are symmetric due to property (ii) of Definition 2. If $j \leq n_l$, consider an arbitrary path from u to a terminal node $\hat{u}$; by construction, there exists some $\hat{\chi} \in \mathcal{T}_{n_l+1,\hat{u}}$ and binaries $(x_j, \dots, x_{n_l})$ such that

$$\hat{\chi} = \chi + \sum_{i=j}^{n_l} \mathbf{a}_i x_i.$$

with $\overline{\phi}(\hat{\chi}) = \nu_{\hat{u}} < +\infty$. This implies that a path starting from u with the same labels $(x_j, \dots, x_{n_l})$ must then also exist from u' to some terminal node $\tilde{u}$, since all $\mathbf{x}$ such that $\hat{\chi} = \mathbf{A}\mathbf{x}$ are represented in $\mathcal{N}$ because they belong to the graph of ϕ, i.e., $\mathscr{G}$. Moreover, $\nu_{\hat{u}} = \overline{\phi}(\hat{\chi}) = \nu_{\tilde{u}}$ by property (ii). That is, the labels of paths from u and u' to the terminal match, as well as the corresponding terminal values. ■ □

Proof of Proposition 5. Let $(\mathbf{x}, \mathbf{y}, z)$ be a feasible solution to the reformulation ($\mathrm{V}^{\mathscr{G}}$). Then, $\phi(\mathbf{x}) < +\infty$ and, by Definition 3-(R) of an approximate value relaxation, together with inequality (3c), we have

$$\mathbf{d}^\top \mathbf{y} \leq z \leq \nu_{u'}$$

for some terminal node $u' \in \mathscr{U}$ associated with $\mathbf{x}$. Thus, there exists some ω (encoding the path associated with $\mathbf{x}$) such that $(\mathbf{x}, \mathbf{y}, z', \omega)$ is also feasible to ($\mathrm{V}^{\mathcal{N}_A}$) for some z', and evaluates to the same objective function value, which only depends on $\mathbf{x}$ and $\mathbf{y}$. Thus, the optimal solution value of ($\mathrm{V}^{\mathcal{N}_A}$) is never worse than v^*. ■ □

Proof of Proposition 6. The formulation (V[$\mathcal{N}_A$]) is equivalent to the system (2),

$$
\begin{aligned}
v_A^* = \min_{\mathbf{x},\mathbf{y}} \quad & \mathbf{c}_x^\top \mathbf{x} + \mathbf{c}_y^\top \mathbf{y} && \text{(2a)}\\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X}(\mathbf{y}) \cap \{0,1\}^{n_l},\ \mathbf{y} \in \mathcal{Y}, && \text{(2b)}\\
& \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \geq \mathbf{h}, && \text{(2c)}\\
& -r\,\mathbf{c}_y^\top \mathbf{y} \leq \phi(\mathbf{x}) + \Delta(\mathbf{x}), && \text{(2d)}
\end{aligned}
$$

where inequality (2d) replaces (8c) and (8d) based on the assumption $\mathbf{d} = -r\,\mathbf{c}_y$ and the definition of $\Delta(\cdot)$. Note that, by definition of $\hat{\mathcal{X}}$, we have $\mathbf{x} \in \hat{\mathcal{X}}$ if and only if there exists $\mathbf{y}$ satisfying (2b)-(2c). Thus, the leader decisions over which (2) is defined are precisely those in $\hat{\mathcal{X}}$. It follows that

$$
v_A^* = \min_{\mathbf{x}\in\hat{\mathcal{X}}} v_A^*(\mathbf{x}).
$$

where

$$
v_A^*(\mathbf{x}) := \min_{\mathbf{y}:\ (\mathbf{x},\mathbf{y})\text{satisfies (2b)-(2d)}} \mathbf{c}_x^\top \mathbf{x} + \mathbf{c}_y^\top \mathbf{y}.
$$

In other words, $v_A^*(\mathbf{x})$ is the minimum objective that can be attained in (2) by minimizing $\mathbf{y}$ for a fixed $\mathbf{x} \in \hat{\mathcal{X}}$, i.e., a leader's decision that is associated with some feasible follower's subproblem. Next, since $r > 0$, rearrange inequality (2d) to

$$
\mathbf{c}_y^\top \mathbf{y} \geq -\frac{1}{r}\phi(\mathbf{x}) - \frac{1}{r}\Delta(\mathbf{x}). \tag{3}
$$

Thus, combining (3) with the definition of $v_A^*(\cdot)$, it follows that

$$
v_A^*(\mathbf{x}) \geq \mathbf{c}_x^\top \mathbf{x} - \frac{1}{r}\phi(\mathbf{x}) - \frac{1}{r}\Delta(\mathbf{x}), \quad \forall \mathbf{x} \in \hat{\mathcal{X}}.
$$

Based on a similar reasoning, the exact formulation (V$^{\mathscr{G}}$) is equivalent to

$$
\begin{aligned}
v^* = \min_{\mathbf{x},\mathbf{y}} \quad & \mathbf{c}_x^\top \mathbf{x} + \mathbf{c}_y^\top \mathbf{y} && \text{(4a)}\\
\text{s.t.} \quad & \text{(2b)-(2c)}, && \text{(4b)}\\
& -r\,\mathbf{c}_y^\top \mathbf{y} \leq \phi(\mathbf{x}). && \text{(4c)}
\end{aligned}
$$

We have that $\phi(\mathbf{x})$ minimizes $-r\,\mathbf{c}_y^\top \mathbf{y}$ for any fixed $\mathbf{x} \in \hat{\mathcal{X}}$. Analogously as above, we have $v^* = \min_{\mathbf{x}\in\hat{\mathcal{X}}} v^*(\mathbf{x})$ for

$$
v^*(\mathbf{x}) := \mathbf{c}_x^\top \mathbf{x} - \frac{1}{r}\phi(\mathbf{x}). \tag{5}
$$

Thus, the difference between bounds for any $\mathbf{x} \in \hat{\mathcal{X}}$ is given by

$$
\begin{aligned}
v^*(\mathbf{x}) - v_A^*(\mathbf{x}) &\leq \mathbf{c}_x^\top \mathbf{x} - \frac{1}{r}\phi(\mathbf{x}) - \mathbf{c}_x^\top \mathbf{x} + \frac{1}{r}\phi(\mathbf{x}) + \frac{1}{r}\Delta(\mathbf{x})\\
&= \frac{1}{r}\Delta(\mathbf{x}).
\end{aligned}
$$

This implies that,

$$\begin{aligned}
v^* - v_A^* &= v^* - \min_{\mathbf{x}\in\hat{\mathcal{X}}} v_A^*(\mathbf{x}) \\
&= v^* + \max_{\mathbf{x}\in\hat{\mathcal{X}}} \{-v_A^*(\mathbf{x})\} \\
&= \max_{\mathbf{x}\in\hat{\mathcal{X}}} \{v^* - v_A^*(\mathbf{x})\} \\
&= \max_{\mathbf{x}\in\hat{\mathcal{X}}} \left\{ \min_{\mathbf{x}'\in\hat{\mathcal{X}}} v^*(\mathbf{x}') - v_A^*(\mathbf{x}) \right\} \\
&\le \max_{\mathbf{x}\in\hat{\mathcal{X}}} \{v^*(\mathbf{x}) - v_A^*(\mathbf{x})\} \\
&\le \frac{1}{r} \max_{\mathbf{x}\in\hat{\mathcal{X}}} \Delta(\mathbf{x}).
\end{aligned}$$

proving the desired lower bound on v_A^*. ■ □

Proof of Proposition 7. Consider the network $\mathcal{N}_A$ prior to the reduction step. Conditions (A) and (B) of Definition 3 are satisfied by construction: each node is assigned to a single layer, and at most two edges labeled in $\{0,1\}$ originate from any node. For condition (R), note that the operations in Step **2**-(b)-(ii) do not eliminate any paths, but redirect multiple paths to terminate at the same node. Thus, the resulting network encodes all binary vectors $\mathbf{x} \in \{0,1\}^{n_l}$.

We show by induction that for any node $u \in \mathscr{U}_j$ and any r–u path labeled $(x_1, \ldots, x_{j-1})$, we have $\chi := \sum_{i=1}^{j-1} \mathbf{a}_i\, x_i \in \mathcal{T}_{j,u}$. This ensures that the terminal values computed in Step **3** yield valid upper bounds, completing the validity argument.

The base case $j = 1$ holds by Step **1**. Suppose the statement holds for all layers $2, \ldots, j-1$, and consider any node $u \in \mathscr{U}_j$ reached by an r–u path labeled $(x_1, \ldots, x_{j-1})$, whose final edge is (u', u, x_{j-1}). By the induction hypothesis, $\chi' := \sum_{i=1}^{j-2} \mathbf{a}_i\, x_i \in \mathcal{T}_{j-1,u'}$. If u is created in Step **2**(a) and not merged in Step **2**(b), then $\chi' + \mathbf{a}_{j-1}\, x_{j-1} \in \mathcal{T}_{j,u}$ by the definition of $\mathcal{T}'$. Otherwise, if u is generated in Step **2**(b), its state set is formed by the union of existing subsets, which preserves all elements from the merged nodes. ■ □

Table 1: Summary of solver results on binary subset of the BOBILib testbed (762 cases).

Method	Feas	Opt	% Gap $\leq$ 1%	% Gap $\leq$ 10%	% Gap $\leq$ 100%	$< z_{\texttt{MibS}}$	$< z_{\texttt{MibS+DD}}$	$< z_{\texttt{B\&C}}$	$< z_{\texttt{B\&C+DD}}$	$< z_{\texttt{DD}}$
MibS	705	0	0.0%	0.1%	2.4%	–	597	277	332	134
MibS + DD	686	2	0.3%	0.4%	45.9%	146	–	53	114	6
B&C	701	14	2.1%	2.9%	99.7%	434	686	–	586	117
B&C + DD	701	6	0.9%	3.4%	90.3%	392	624	118	–	40
DD	751	137	21.3%	24.1%	99.9%	619	744	616	707	–

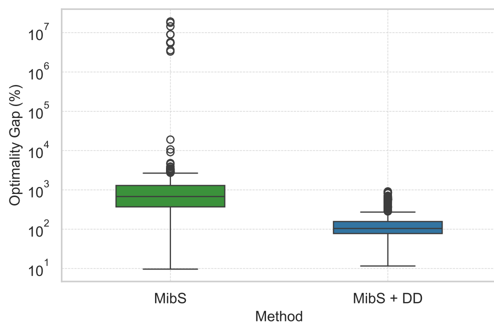
B Additional Results on Solver Integrations

We next discuss initial results obtained by incorporating value networks into the two state-of-the-art solvers, MibS and B&C, as well as the opportunities and challenges that this integration raises for future research.

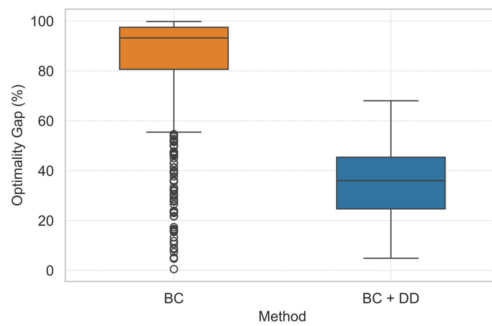
Setup and benchmark. For this analysis, we extracted the value networks $\mathcal{N}_A$ constructed by the DD method with budget $B = 8$, built the extended formulation $\mathrm{V}[\mathcal{N}_A]$, and generated the updated MPS and AUX input files required by MibS and B&C. Both solvers were then run with the same time limit of 7,200 seconds used in Section 6. This represents an idealized improvement scenario, since the value-network construction itself is assumed to incur zero compilation time. Due to possible numerical issues arising when converting files to MPS, we restrict attention to the subset of 762 instances from the testbed described in §6.1 for which both leader and follower variables are binary, and therefore all costs are integral. The augmented methods are denoted by MibS+DD and B&C+DD, respectively.

Results. Table 1 and the optimality-gap boxplots in Figure 1 summarize the results. Overall, both suggest that the network information can materially improve the quality of the bounds produced by MibS and B&C, but the effect is solver dependent and comes with nontrivial trade-offs in feasibility and proven optimality. For MibS, incorporating the network substantially reduces the optimality gaps among unsolved instances. As shown in Figure 1a, the distribution shifts downward by roughly an order of magnitude on the log scale: the median gap decreases from 678.3% for MibS to 104.2% for MibS+DD, and the upper tail is also compressed, with the 75th percentile dropping from 1296.0% to 156.8%. This improvement is also reflected in Table 1, where the proportion of feasible instances with gap at most 100% rises sharply from 2.4% to 45.9%. Nonetheless, this strengthening does not translate into a substantial increase in proven optimality, since MibS+DD solves only 2 instances to optimality, versus none for MibS, and it reduces the number of feasible solutions found from 705 to 686. Thus, for MibS, the main benefit of value networks is not improved robustness in feasibility or exact completion, but rather a marked improvement in bound quality.

The behavior is distinct for B&C. Figure 1b shows that adding the network again shifts the gap distribution downward, with the mean gap decreasing from 84.4% to 35.0% and the median from 93.2% to 36.0%. The bulk of the distribution therefore moves away from the high-gap regime that dominates B&C alone. However, unlike in the MibS case, B&C already performs strongly on this testbed in terms of feasibility and moderate-to-good gaps. In particular, B&C attains feasibility on 701 instances and proves optimality on 14, while B&C+DD preserves the same number of feasible instances but reduces the number of proven optima to 6. Moreover, the percentage of instances with gap at most 100% decreases from 99.7% to 90.3%, even though the average and median gaps improve. This indicates that the value network helps substantially on the central mass of difficult instances, but may also worsen performance on a subset of cases where the original branch-and-cut procedure of Fischetti et al. (2017) already performs well.



(a) `MibS` vs. `MibS+DD` (log-scale)



(b) `B&C` vs. `B&C+DD`

Figure 1: (Colored.) Optimality gap comparison on binary subset of the BIBILib testbed (762 cases)

Opportunities and challenges. These results suggest that the information from value networks is complementary to existing generic bilevel solvers and their underlying methodologies, but not a universal plug-in improvement. For `MibS`, the combination yields a clear gain in gap quality, driven by the stronger lower bounds provided by the value-network formulation on this benchmark, albeit with some loss in feasibility and only limited gains in exact solves. For `B&C`, the combination improves the central tendency of the gaps, but weakens some of the strong tail performance and exact completions of the original solver.

These experiments suggest that the value-network formulation interacts with the cut generation and branch-and-bound methodologies of each solver in ways that require future research and development. In particular, adding the extended formulation may change which cuts are generated, how effective they are, and how the search tree evolves. A more careful study is therefore needed to identify which classes of cuts remain useful in the presence of value networks, which become redundant, and how branch-and-bound procedures should be adapted to exploit the additional network structure more effectively.

References

Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ.

Fischetti, Matteo, Ivana Ljubić, Michele Monaci, Markus Sinnl. 2017. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research* **65**(6) 1615–1637.