Forbidden Vertices for some classes of 0 - 1 polytopes

Esteban Salgado\textsuperscript{1,2}, Gustavo Angulo\textsuperscript{3}

\textsuperscript{1} Istituto di Analisi dei Sistemi ed Informatica “Antonio Ruberti” – CNR, Italy, esteban.salgado@iasi.cnr.it
\textsuperscript{2} Department of Computer, Control and Management Engineering Antonio Ruberti, Sapienza University of Rome, Italy
\textsuperscript{3} Department of Industrial and Systems Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile gangulo@ing.puc.cl

Abstract

The forbidden-vertices problem aims to optimize a linear function over the vertices of a polytope that remains after prohibiting a given subset of them. We present extended formulations for this problem for some classes of $0 - 1$ polytopes. The sizes of the formulations are smaller than the known bounds for these polytopes. We also compare this formulation on the Prize collecting TSP with other approaches that describe the same integer set, but generate different polytopes.

Keywords: Forbidden vertices, extended formulation.

1 Introduction

Given an optimization problem $\mathcal{P} = \min \{c^\top x | x \in X\}$ defined over the polytope $X \subseteq \mathbb{R}^n$ whose vertices are $\text{vert}(X) \subseteq \{0,1\}^n$, it could be needed to optimize without considering a subset of the vertices $V \subseteq \text{vert}(X)$. For instance, the problem of finding the $\kappa$-best solutions of $\mathcal{P}$ gets into this framework by solving and dynamically adding the optimal solution to $V$ until $|V| = \kappa$. Other examples come from enumerative schemes for stochastic integer programming, such as the integer L-shaped method [3], in which a subset of the solution set is known and optimization over the remaining feasible solutions is carried out.

The most naive approach to remove a set $V$ of binary vertices is to add a no-good-cut constraint for each $v \in V$, i.e. a constraint that only cuts the vertex $v$. These $|V|$ constraints act independently and therefore rise doubts about the strength of the formulation obtained.

Motivated by this, [2] introduced the problem of forbidden-vertices stated as follows: Given a polytope $X \subseteq \mathbb{R}^n$, a set $V \subseteq \text{vert}(X)$, and a vector $c \in \mathbb{R}^n$, the forbidden-vertices problem is to either assert $\text{vert}(X) \setminus V = \emptyset$, or to return an element in $\arg\min \{c^\top x | x \in \text{vert}(X) \setminus V\}$.

It is known that when $X = [0,1]^n$, it holds that $xc(\text{forb}(X,V)) \in \mathcal{O}(n|V|)$ [2], where $\text{forb}(X,V) := \text{conv}(\text{vert}(X) \setminus V)$ and $xc(P)$ is the extension complexity of $P$, i.e. the minimal number of facets in an extension of $P$. They also proved that when $X$ is a general $0 - 1$ polytope, it holds that $xc(\text{forb}(X,V)) \in \mathcal{O}(n|V|(xc(X) + 1))$.

In this work we present extended formulations when $X$ is:

\begin{itemize}
  \item $X_k^\leq = \{x \in [0,1]^n | 1^\top x \leq k\}$. In the following we will call it “$(\leq k)$-Simplex”
  \item $X_G^s-t = \{x \in [0,1]^{E(G)} | x \text{ is a } s-t \text{ path in } G\}$ where $G$ is a DAG. In the following we will call it “$s-t$ path polytope”
\end{itemize}

Our extended formulations yield smaller bound on the extension complexity than those in [2].
2 Extended formulations

In the two cases that we will analyze the procedure will be the same. Given a set \( V \subseteq \text{vert}(X) \) of non-valid vertices, we will define a set \( W \) of prefixes, i.e. vectors of length \( i \leq n \) such that any completion \( y \in \{0, 1\}^{n-i} \) of the remaining \((n-i)\) components feasible for \( \text{vert}(X) \), satisfies \((w, y) \in \text{vert}(X) \setminus V\). Then we will optimize over the remaining components, finding the best selection of a prefix with its completion.

The set of prefixes can be easily computed. If \((V^i, X^i)_{1 \leq i \leq n}\) are the projections of \( V \) and \( X \) onto the first \( i \) components, respectively, then \( W \) can be computed as \( W = (X^1 \setminus V^1) \cup \bigcup_{i=2}^{n} ([V^{i-1} \times \{0, 1\}] \cap X^i) \setminus V^i\).

2.1 \((\leq k)\)-Simplex

Let \( W_{ib} \) be the set of prefixes \( w \) of length \( i \) with \( 1^\top w = n - b \), where \( T \) is the set of the pairs \((i,b)\) such that \( W_{ib} \neq \emptyset \). The optimal value of forbidden-vertices problem can be found by solving

\[
\gamma = \min_{(i,b) \in T} \left\{ \min_{w \in W_{ib}} \sum_{j=1}^{i} c_j w_j + \min_{x \in X_{n-i}^b} \sum_{j=i+1}^{n} c_j x_{j-i} \right\},
\]

In other words, \( \alpha_{ib} \) is the objective value of the best prefix with \( i \) entries and \((n-b)\) ones, while \( \beta_{ib} \) is the objective value of the best completion, which is obtained by optimizing over a \((\leq b)\)-Simplex in dimension \( n - i \).

Note that if \( n < i \), then \( \beta_{ib} = \min\{\beta_{i+1}b, c_{i+1} + \beta_{i+1}(b-1)\} \) if \( b > 1 \). Therefore, \( \gamma \) can be computed by dynamic programming (DP), which can be equivalently formulated as the LP \( \max\{\gamma \mid (1)\} \) with unrestricted variables \( \alpha, \beta, \gamma \) where (1) is defined by

\[
\begin{align*}
\gamma &\leq \alpha_{ib} + \beta_{ib} \quad (\forall (i,b) \in T) \quad (1a) \\
\alpha_{ib} &\leq \sum_{j=1}^{i} c_j w_{ib}^{jh} \quad (\forall (i,b) \in T, h \leq \ell_{ib}) \quad (1b) \\
\beta_{ib} &\leq \beta_{i+1}b \quad (\forall 1 \leq i \leq n-1, 0 \leq b \leq k) \quad (1c) \\
\beta_{ib} &\leq c_{i+1} + \beta_{i+1}(b-1) \quad (\forall 1 \leq i \leq n-1, 1 \leq b \leq k) \quad (1d) \\
\beta_{ib} &\geq 0 \quad (\forall 0 \leq b \leq k) \quad (1e) \\
\beta_{ib} &\geq 0 \quad (\forall 1 \leq i \leq n) \quad (1f)
\end{align*}
\]

where the variables \( \varphi, \pi, \sigma, \eta, \delta, \varepsilon \) are the corresponding dual variables.

By strong duality of LP, the dual has the same optimal value. The dual is to minimize

\[
\sum_{j=1}^{n} c_j \left[ \sum_{(i,b) \in T} w_{ib} \pi_{ib} + \chi(2 \leq j) \sum_{1 \leq b \leq k} \eta_{(j-1)b} \right] \text{ subject to }
\]

\[
\begin{align*}
\varphi_{ib} - \sum_{1 \leq b \leq |W_{ib}|} \pi_{ib} &= 0 \quad (\forall (i,b) \in T) \quad (2a) \\
\sum_{(i,b) \in T} \varphi_{ib} &= 1 \quad (2b) \\
F_{ib}(\varphi, \pi, \sigma, \eta, \delta, \varepsilon) &= 0 \quad (\forall 1 \leq i \leq n, 0 \leq b \leq k) \quad (2c) \\
\varphi, \pi, \sigma, \eta &\geq 0 \quad (2d)
\end{align*}
\]
where $\chi_A$ is the indicator function of proposition $A$ and $F_{ib}(\varphi, \pi, \sigma, \eta, \delta, \varepsilon) = \sigma(i-1)b\chi(2\leq i) - \sigma_{ib}\chi(i<n) + \eta(i-1)(b+1)\lambda(2\leq i, 1\leq k) - \eta_{ib}\chi(i<n, 1\leq k) + \varphi_{ib}\chi(i, b) + \delta_{ib}\chi(i=n) + \varepsilon_{ib}\lambda(b=0)$. From duality, we see that (2) is the original problem we wanted to solve. Therefore, (2a) – (2d) coupled with

$$x_j = \sum_{1 \leq i \leq m} w_{ib}^j \pi_{ibh} + \chi(2\leq j) \sum_{1 \leq b \leq k} \eta(j-1)b \quad (\forall 1 \leq j \leq n)$$

defines an extended formulation for $\text{forb}(X^n_k, V)$ of size $O(n^2 + n|V|)$, which is less than the known bound of $O(n^2|V| + n|V|)$ [2].

2.2 $s - t$ path polytope

Wlog, we can always consider a DAG $G = (N, A)$ where every node and arc is in some $s - t$ path, with nodes sorted in a topological order $\{s = v_1, \ldots, v_n = t\}$ and arcs sorted in lexicographic order. Let $W_{iv}$ be the set of prefixes of length $i$ that describe an $s - v$ path and $T$ the set of all $(i, v)$ such that $W_{iv} \neq \emptyset$. We define the DP

$$\gamma = \min_{(i,v) \in T} \left\{ \min_{w \in W_{iv}} \sum_{j=1}^{i} c_j w_j + \min_{x \in X_{i+1}} \sum_{j=i+1}^{m} c_j x_j \right\}$$

Proceeding analogously to the previous case, we get an extended formulation of size $O(|A||V| + |A|^2)$, which is better than the known bound $O(|A||V| + |A|^2|V|)$ [2].

Given that several DPs can be transformed into the problem of finding an $s - t$ path in a DAG, this formulation can be used to solve the forbidden-vertices over a larger class of problems.

3 Computational results

To test the performance of the first formulation, we consider the Prize collecting TSP (PCTSP) in which the objective is to find an optimal cycle of length at most $k$ in a complete directed graph with $n$ nodes, with a depot node in which the cycle must start, with costs for using an edge and costs for not using a node. Therefore the problem can be described as an IP

$$\min \{ c^T x + d^T y \mid (x, y) \in P \cap \{0, 1\}^{n(n-1)} \times \text{vert}(X^n_{n-k}) \}$$

and we can prohibit combinations of selected nodes $V$. We compare three formulations:

1. $\min \{ c^T x + d^T y \mid (x, y) \in P \cap \{0, 1\}^{n(n-1)} \times \text{forb}(X^n_{n-k}, V) \}$ with the extended formulation described in section 2.1.

2. $\min \{ c^T x + d^T y \mid (x, y) \in P \cap \{0, 1\}^{n(n-1)} \times \text{vert}(X^n_{n-k}) \cap N(V) \}$ where $N(V)$ is described by the constraints $\sum_{i:v_i=1} (1 - y_i) + \sum_{i:v_i=0} y_i \geq 1$ for every $v \in V$. These are the no-good-cut constraints.

3. $\min \{ c^T x + d^T y \mid (x, y) \in P \cap \{0, 1\}^{n(n-1)} \times \text{forb}([0, 1]^n, V) \cap X^n_{n-k} \}$ with the extended formulation described in [1].

For several values of $n$ and $k \in [0, [n/4], [n/2], [3n/4], n - 1]$, we obtained the $\kappa$-best solutions for the PCTSP with $\kappa = 5$ and computed the time needed by the solver to solve the LP relaxation (time rel) and the MILP (time opt), and also its gap of optimality (gap) in percentage. The test was performed using Gurobi 9.0.1 with one thread. We summarize the results and present a subset of them in table 1.

The marked values are the best for each configuration of $(n, k)$ averaged on the $\kappa$ iterations of the problem. The process was repeated on 20 different instances by configuration. In the
Instance | No-good-cut
| forb([0, 1]^n, \mathbb{V}) \cap X_{n-k}^{n-k} | forb(X_{n-k}^{n-k}, \mathbb{V})
<table>
<thead>
<tr>
<th>N</th>
<th>k</th>
<th>gap(%)</th>
<th>time opt(s)</th>
<th>time rel(s)</th>
<th>gap(%)</th>
<th>time opt(s)</th>
<th>time rel(s)</th>
<th>gap(%)</th>
<th>time opt(s)</th>
<th>time rel(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.43</td>
<td>69.964</td>
<td>19.772</td>
<td>1.31</td>
<td>73.604</td>
<td>21.109</td>
<td>1.31</td>
<td>72.067</td>
<td>21.067</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.43</td>
<td>81.768</td>
<td>18.721</td>
<td>1.31</td>
<td>83.906</td>
<td>20.648</td>
<td>1.31</td>
<td>85.506</td>
<td>19.994</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>20</td>
<td>1.42</td>
<td>57.983</td>
<td>19.013</td>
<td>1.31</td>
<td>67.878</td>
<td>20.621</td>
<td>1.30</td>
<td>78.654</td>
<td>20.639</td>
</tr>
<tr>
<td>30</td>
<td>1.43</td>
<td>98.193</td>
<td>21.071</td>
<td>1.31</td>
<td>78.998</td>
<td>19.053</td>
<td>1.31</td>
<td>64.777</td>
<td>20.961</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>1.42</td>
<td>115.945</td>
<td>24.589</td>
<td>1.31</td>
<td>105.057</td>
<td>24.092</td>
<td>1.31</td>
<td>88.616</td>
<td>22.154</td>
<td></td>
</tr>
</tbody>
</table>

TAB. 1: Comparative results for the three formulations.

The case of \textit{gap}, there are differences between the values that look equal of magnitude less than $10^{-5}$.

The results show that the formulation presented in section 2.1 outperforms the other two when $n - k$ is smaller. This intuitively can be explained because when $n - k$ is closer to $n$ the polytope $X_{n-k}^{n}$ is more similar to $[0, 1]^n$, therefore a description of the hypercube with fewer variables and constraints is faster and effective. We also see that the optimality gap of no-good-cut is always dominated by the forbidden-vertices approaches and the election between them is based on the value $n - k$.

4 Conclusions and perspectives

Given that solving some DPs can be transformed into finding a $s-t$ path in a DAG, the formulation presented in section 2.2 might have a wide set of problems in which it can be used. Nevertheless, usually the prohibited vertices will come in the original space, thus finding a way to relate this forbidden vertices into the transformed space is needed. Also, in [4] it was shown a polyhedral characterization of discrete dynamic programming using a definition of directed acyclic hypergraph. Therefore extending this procedure into that framework is also an interesting research direction.

References


