Forbidden Vertices for some classes of 0 - 1 polytopes

Esteban Salgado[1,2], Gustavo Angulo[3]

[1] Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" – CNR, Italy,
esteban.salgado@iasi.cnr.it

[2] Department of Computer, Control and Management Engineering Antonio Ruberti, Sapienza University of Rome, Italy

[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 $\mathcal{P} = \min\{c^\top x \mid x \in X\}$ defined over the polytope $X \subseteq \mathbb{R}^n$ whose vertices are vert$(X) \subseteq \{0,1\}^n$, it could be needed to optimize without considering a subset of the vertices $V \subseteq$ vert(X). For instance, the problem of finding the κ-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 $V \subseteq \mathtt{vert}(X)$, and a vector $c \in \mathbb{R}^n$, the *forbidden-vertices* problem is to either assert $\mathtt{vert}(X) \setminus V = \emptyset$, or to return an element in $\mathtt{argmin}\{c^\top x \mid x \in \mathtt{vert}(X) \setminus V\}$.

It is known that when $X = [0,1]^n$, it holds that $\mathtt{xc}(\mathtt{forb}(X,V)) \in \mathcal{O}(n|V|)$ [2], where $\mathtt{forb}(X,V) := \mathtt{conv}(\mathtt{vert}(X) \setminus V)$ and $\mathtt{xc}(P)$ 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 $\mathtt{xc}(\mathtt{forb}(X,V)) \in \mathcal{O}(n|V|(\mathtt{xc}(X)+1))$.

In this work we present extended formulations when X is:

- $X_k^n = \{x \in [0,1]^n \mid 1^\top x \le k\}$. In the following we will call it "$(\le k)-$Simplex"
- $X_{st}^G = \{x \in [0,1]^{E(G)} \mid x \text{ is a } s-t \text{ path in } G\}$ where G is a DAG. In the following we will call it "$s-t$ path polytope"

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 $V \subseteq \texttt{vert}(X)$ 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 $\texttt{vert}(X)$, satisfies $(w,y) \in \texttt{vert}(X) \setminus V$. 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 $(\leq \mathbf{k})-$Simplex

Let W_{ib} be the set of prefixes w of length i with $1^\top w = n - b$, where $\mathcal{T}$ 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

$$\gamma = \min_{(i,b)\in\mathcal{T}} \left\{ \underbrace{\min_{w\in W_{ib}} \sum_{j=1}^{i} c_j w_j}_{=:\alpha_{ib}} + \underbrace{\min_{x\in X_b^{n-i}} \sum_{j=i+1}^{n} c_j x_{j-i}}_{=:\beta_{ib}} \right\}.$$

In other words, α_{ib} is the objective value of the best prefix with i entries and $(n-b)$ ones, while β_{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, γ can be computed by dynamic programming (DP), which can be equivalently formulated as the LP $\max\{\gamma \,|\, (1)\}$ with unrestricted variables α, β, γ where (1) is defined by

$$\gamma \leq \alpha_{ib} + \beta_{ib} \qquad (\forall\,(i,b)\in\mathcal{T}) \quad (\varphi_{ib}) \tag{1a}$$

$$\alpha_{ib} \leq \sum_{j=1}^{i} c_j w_j^{ibh} \qquad (\forall\,(i,b)\in\mathcal{T},\ h \leq \ell_{ib}) \quad (\pi_{ibh}) \tag{1b}$$

$$\beta_{ib} \leq \beta_{(i+1)b} \qquad (\forall\, 1\leq i\leq n-1,\ 0\leq b\leq k) \quad (\sigma_{ib}) \tag{1c}$$

$$\beta_{ib} \leq c_{(i+1)} + \beta_{(i+1)(b-1)} \qquad (\forall\, 1\leq i\leq n-1,\ 1\leq b\leq k) \quad (\eta_{ib}) \tag{1d}$$

$$\beta_{nb} = 0 \qquad (\forall\, 0\leq b\leq k) \quad (\delta_{nb}) \tag{1e}$$

$$\beta_{i0} = 0 \qquad (\forall\, 1\leq i\leq n) \quad (\varepsilon_{i0}) \tag{1f}$$

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 $\sum_{j=1}^{n} c_j \left[\sum_{\substack{(i,b)\in\mathcal{T} \\ 1\leq h\leq|W_{ib}|,1\leq j\leq i}} w_j^{ibh}\pi_{ibh} + \chi_{(2\leq j)} \sum_{1\leq b\leq k} \eta_{(j-1)b} \right]$ subject to

$$\varphi_{ib} - \sum_{1\leq h\leq|W_{ib}|} \pi_{ibh} = 0 \qquad (\forall (i,b)\in\mathcal{T}) \tag{2a}$$

$$\sum_{(i,b)\in\mathcal{T}} \varphi_{ib} = 1 \tag{2b}$$

$$F_{ib}(\varphi,\pi,\sigma,\eta,\delta,\varepsilon) = 0 \qquad (\forall 1\leq i\leq n,\ 0\leq b\leq k) \tag{2c}$$

$$\varphi,\pi,\sigma,\eta \geq 0 \tag{2d}$$

where χ_A is the indicator function of proposition A and $F_{ib}(\varphi, \pi, \sigma, \eta, \delta, \varepsilon) = \sigma_{(i-1)b}\chi_{(2\leq i)} - \sigma_{ib}\chi_{(i<n)} + \eta_{(i-1)(b+1)}\chi_{(2\leq i, b<k)} - \eta_{ib}\chi_{(i<n, 1\leq b)} + \varphi_{ib}\chi_{((i,b)\in\mathcal{T})} + \delta_{ib}\chi_{(i=n)} + \varepsilon_{ib}\chi_{(b=0)}$. From duality, we see that (2) is the original problem we wanted to solve. Therefore, (2a) – (2d) coupled with

$$x_j = \sum_{\substack{(i,b)\in\mathcal{T} \\ 1\leq h\leq |W_{ib}|, 1\leq j\leq i}} w_j^{ibh}\pi_{ibh} + \chi_{(2\leq j)} \sum_{1\leq b\leq k} \eta_{(j-1)b} \ (\forall 1 \leq j \leq n)$$

defines an extended formulation for $\texttt{forb}(X_k^n, V)$ of size $\mathcal{O}(n^2 + n|V|)$, which is less than the known bound of $\mathcal{O}(n^2|V| + n|V|)$ [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 $\mathcal{T}$ the set of all (i, v) such that $W_{iv} \neq \emptyset$. We define the DP

$$\gamma = \min_{(i,v)\in\mathcal{T}} \left\{ \underbrace{\min_{w\in W_{iv}} \sum_{j=1}^{i} c_j w_j}_{=:\alpha_{iv}} + \underbrace{\min_{x\in X_{v,t}} \sum_{j=i+1}^{m} c_j x_{j-i}}_{=:\beta_{iv}} \right\}$$

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

Given that several *DP*s 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 $\min\{c^\top x + d^\top y \,|\, (x, y) \in P \cap (\{0, 1\}^{n(n-1)} \times \texttt{vert}(X_{n-k}^n))\}$ and we can prohibit combinations of selected nodes V. We compare three formulations:

1. $\min\{c^\top x + d^\top y \,|\, (x, y) \in P \cap [\{0, 1\}^{n(n-1)} \times \texttt{forb}(X_{n-k}^n, V)]\}$ with the extended formulation described in section 2.1.

2. $\min\{c^\top x + d^\top y \,|\, (x, y) \in P \cap [\{0, 1\}^{n(n-1)} \times (\texttt{vert}(X_{n-k}^n) \cap N(V))]\}$ where $N(V)$ is described by the constraints $\sum\limits_{i:v_i=1} (1 - y_i) + \sum\limits_{i:v_i=0} y_i \geq 1$ for every $v \in V$. These are the *no-good-cut* constraints.

3. $\min\{c^\top x + d^\top y \,|\, (x, y) \in P \cap [\{0, 1\}^{n(n-1)} \times (\texttt{forb}([0, 1]^n, V) \cap X_{n-k}^n)]\}$ with the extended formulation described in [1].

For several values of n and $k \in [0, \lfloor n/4\rfloor, \lfloor n/2\rfloor, \lfloor 3n/4\rfloor, n - 1]$, we obtained the κ-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 κ iterations of the problem. The process was repeated on 20 different instances by configuration. In the

Instance		No-good-cut			$\texttt{forb}([0,1]^n, V) \cap X^n_{n-k}$			$\texttt{forb}(X^n_{n-k}, V)$		
N	k	gap(%)	time opt(s)	time rel(s)	gap(%)	time opt(s)	time rel(s)	gap(%)	time opt(s)	time rel(s)
40	0	1.43	**69.964**	**19.772**	**1.31**	73.604	21.109	**1.31**	72.067	21.067
	10	1.43	**81.768**	**18.721**	**1.31**	83.906	20.648	**1.31**	85.506	19.994
	20	1.42	**57.983**	**19.013**	1.31	67.878	20.621	**1.30**	78.654	20.639
	30	1.43	98.193	21.071	**1.31**	78.998	**19.053**	**1.31**	**64.777**	20.961
	39	1.42	115.945	24.589	**1.31**	105.057	24.092	**1.31**	**88.616**	**22.154**
50	0	2.08	**217.627**	101.232	**2.02**	239.063	**91.142**	**2.02**	244.419	91.328
	12	2.08	**228.860**	90.088	**2.02**	236.479	88.152	**2.02**	229.157	**84.561**
	25	2.08	219.868	96.456	**2.02**	218.359	91.988	**2.02**	**216.824**	**89.751**
	37	2.08	274.350	94.180	**2.02**	246.844	**91.091**	**2.02**	**219.868**	96.456
	49	2.08	250.276	93.379	**2.02**	264.533	91.729	**2.02**	**249.530**	**90.112**

TAB. 1: Comparative results for the three formulations.

case of `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_{n-k} 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

[1] Gustavo Angulo, Shabbir Ahmed, and Santanu S. Dey. Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

[2] Gustavo Angulo, Shabbir Ahmed, Santanu S. Dey, and Volker Kaibel. Forbidden vertices. *Mathematics of Operations Research*, 40(2):350–360, 2015.

[3] Gilbert Laporte and François V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133 – 142, 1993.

[4] R. Kipp Martin, Ronald L. Rardin, and Brian A. Campbell. Polyhedral characterization of discrete dynamic programming. *Operations Research*, 38(1):127–138, 1990.