Extracting Alternative Solutions from Benders Decomposition

Matthew Viens*[1,2], William E. Hart†[2] and Michael Ferris‡[1]

[1]Department of Computer Sciences, University of Wisconsin-Madison
[2]Center for Computing Research, Sandia National Laboratories

September 2025

Abstract

We show how to extract alternative solutions for optimization problems solved by Benders Decomposition. In practice, alternative solutions provide useful insights for complex applications; some solvers do support generation of alternative solutions but none appear to support such generation when using Benders Decomposition. We propose a new post-processing method that extracts multiple optimal and near-optimal solutions using the cut-pool generated during Benders Decomposition. Further, we provide a geometric framework for understanding how the adaptive approximation in Benders Decomposition relates to alternative solutions. We demonstrate this technique on stochastic programming and interdiction modeling, and we highlight use cases that require the ability to enumerate all optimal solutions.

1 Introduction

Optimization solvers traditionally return a single optimal solution. We know that multiple solutions naturally occur in a variety of applications, but computational tools to generate alternative solutions remain limited. However, alternative solutions often provide practical value. For example, end-users may have secondary concerns, like unexpressed objectives or modeling uncertainty, that motivate an analysis of alternative solutions (Brill 1979). There is a growing body of literature that describes methods to generate and diversify alternative solutions (e.g. Lau et al. 2024, Petit and Trapp 2019, Ahanor et al. 2024), and commercial solvers have begun to integrate this functionality (e.g. Gurobi, CPLEX). However, previous work has not considered decomposition methods that generate alternative solutions. This treatment of more specific algorithms has not been helped by the scattered and diverse names used to describe several solutions to an optimization problem, especially in treating both exact optimal or near optimal solutions; we standardize on the term alternative solutions for this concept. We show how to adapt Benders Decomposition (Benders 1962, Van Slyke and Wets 1969) to identify alternative solutions. We focus on, and we describe stochastic programming and interdiction applications for our alternative solution exemplars.

*mviens@wisc.edu
†wehart@sandia.gov
‡ferris@cs.wisc.edu

Funding: This work was supported in part by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. Additional funding came from the Morgridge Chair of Computer Sciences at University of Wisconsin-Madison. This research was performed while some authors were participating in the Architecture of Green Energy Systems Program hosted by the University of Chicago, which is supported by the National Science Foundation (Grant No. DMS-1929348).

Acknowledgements: We would like to express our sincere gratitude to Cynthia Phillips for her invaluable contributions to this research. We are also grateful to Jared Gearhart for their support and assistance throughout the course of this work.

We propose a new post-processing method that extracts alternative optimal and near-optimal solutions using the cut-pool generated during Benders Decomposition. Our overall approach involves three steps. We solve a Benders Decomposition problem to optimality. Then we use the generated cut-pool in our post-processing method to generate candidate alternative solutions. Finally, we filter the candidate alternative solutions with a certification step. We call this approach AOS-Benders (alternative optimal solutions for Benders).

We motivate AOS-Benders by describing an epigrapical and level-set view of the connection between alternative solutions in the original problem and the Benders Decomposition problem. Benders Decomposition separates into two problems (master and subproblem) that have a corresponding split in variables matching the first-stage and second-stage variables from stochastic programming; we use this first-stage and second-stage variable convention to describe our variable distinction. We describe two core theoretical results. First, the set of alternative solutions for the Benders master problem contains the set of alternative solutions for the first-stage variables. Second, the converse is generally not true. Alternative solutions in the second-stage variables are also possible. Our framework also treats the differing requirements of generating exact and approximate alternative solutions, and it provides methods for generating both kinds.

The paper is laid out as follows. In Section 2, we review alternative solution theory and generation methods. In Section 3, we review Benders Decomposition. We develop a theoretical analysis and describe AOS-Benders in Section 4. We present the Farmer's Problem as a stochastic programming exemplar in Section 5, and we present the s-t Shortest Path Problem as an interdiction exemplar in Section 6. We conclude with a discussion of results and future work in Section 7. For formatting of variables and data, we use uppercase calligraphic for matrices, uppercase italics for sets, and lowercase (bold) italics for (vector) data/variables. For formatting of operators, we use roman and bold when vectorized.

2 Alternative Solutions

We define 'alternative solutions' as a term that relates to the generation of several solutions to an optimization problem. There are many contexts where alternative solutions have been treated under differing names including 'alternative optimal solutions' by Paris (2010) and Williams (2013), 'multiple optimal solutions' also by Paris (1981), 'modeling to generate alternatives' by Brill et al. (1982, 1990), 'complete local minimizer (CLM)' sets by Robinson (1996), 'set of all optimal solutions' by Rockafellar (1996), and the 'set of minimizing points' by Bertsekas et al. (2013). The different names correspond to research communities that do not seem to interact. Rockafellar and Bertsekas discuss alternative solutions in existence arguments for solution sets. Robinson treats alternative solutions in sample path optimization for simulation for a discussion about set compactness and connectedness. Paris, Williams, and Brill each treat alternative solutions in discussions for their generation. Even in the generation discussions, the different terms date back to different disciplines. The Management Sciences literature starts with Brill (1979) and the Agricultural Economics literature starts with Paris (1981). While both areas continued to develop new generation methods (e.g. Brill et al. 1982, Paris 1983b) and debate use cases for alternative solutions (e.g. Brill et al. 1990, Paris 1983a), there does not appear to be a standard notation or representation for alternative solutions either in the early literature or more recent application-specific reviews (e.g. Lau et al. 2024).

Since we are concerned both about theory and generation, our presentation divides naturally into these two parts. In Section 2.1, we describe a mathematical framework for alternative optimal and near-optimal solutions based on sublevel sets. We use sublevel sets to describe the differences between alternative solutions to the Benders master problem and alternative solutions to the extensive-form problem in Section 4.1. In Section 2.2, we review previous research and available software to generate alternative solutions in a range of problem types. We leverage these methods to generate alternative solutions for the first-stage variables in Section 4.2 and for the second-stage variables in Section 4.3.

2.1 Alternative Solutions Theory

We adapt a mathematical framework to describe alternative solutions for both exact optimal solutions and near optimal solutions from the existence arguments of Bertsekas and Robinson. Consider the following optimization problem:

$$z^* = \min_{\boldsymbol{x} \in X} \mathrm{f}(\boldsymbol{x}),$$

where f is an objective function defined over feasible domain X. The set of *exact alternative optimal solutions* (EX-ALT) is the level set:

$$\begin{aligned} \mathrm{S}(\mathrm{f}, X, z^*) &= \{\boldsymbol{x} \in X \mid \mathrm{f}(x) = z^*\} \\ &= \{\boldsymbol{x} \in X \mid \mathrm{f}(x) \leq z^*\}. \end{aligned}$$

These are equivalent definitions for this level set since $\{\boldsymbol{x} \in X \mid \mathrm{f}(\boldsymbol{x}) < z^*\} = \varnothing$, but the second definition is the sublevel set with level z^*. Sublevel sets are convex for quasi-convex functions, which includes convex functions (see e.g. Bertsekas et al. 2013). Further, we consider a general level value of τ rather than z^*:

$$\mathrm{S}(\mathrm{f}, X, \tau) = \{\boldsymbol{x} \in X \mid \mathrm{f}(\boldsymbol{x}) \leq \tau\}.$$

When the level value $\tau > z^*$, this is the set of *approximate alternative optimal solutions* (A-ALT). We can use this for an absolute or relative tolerance from z^* by choosing $\tau = z^* + \epsilon$ or $\tau = (1 + \alpha)z^*$ respectively.

2.2 Generation of Alternative Solutions

Previous research has developed methods to extract alternative solutions from both Linear Programs (LPs) and Mixed-Integer Programs (MIPs) for both A-ALT and EX-ALT sets. We draw a distinction between black-box and white-box generation techniques. The black-box techniques are alternative solution generation methods where the implementation details are unknown. White-box techniques for LPs include iterative MIP methods (Lee et al. 2000) and simultaneous discovery methods (Pedersen et al. 2021). Approaches for MIPs include No-Good Cuts methods for 0-1 problems (Balas and Jeroslow 1972) and heuristic methods like keeping track of the N best incumbents in branch-and-bound (Eckstein et al. 2015). For black-box methods, both Gurobi and CPLEX provide ways of generating alternative solutions as part of their MIP solvers in solution pool structures. Though the solvers provide guarantees, both note challenges when generating alternative solutions for MIPs that mix continuous and discrete variables (Gurobi Optimization, LLC 2024, IBM ILOG CPLEX 2024). Modeling languages like AIMMS, GurobiPy, and Pyomo can generate and represent alternative solutions. AIMMS can use CPLEX's Solution Pool (AIMMS 2024), and GurobiPy uses Gurobi's Solution Pool (Gurobi Optimization, LLC 2024). Pyomo (Bynum et al. 2021) can use Gurobi's solution pool, and it includes solver-agnostic methods for generating alternative solutions with non-commercial solvers like GLPK and HiGHS (Hart et al. 2024). In our experiments in Sections 5 and 6, we use the Pyomo generation methods for what Algorithm 2 calls an AOSKernel. For linear programs, we use the `enumerate_linear_solutions` method (hereafter Pyomo-AOS-Linear), which implements a version of the vertex-enumeration strategy from Lee et al. (2000). For 0-1 problems, we use the `enumerate_binary_solutions` method (hereafter Pyomo-AOS-Binary), which implements a version of No-Good Cuts method from Balas and Jeroslow (1972). Both methods can exhaustively enumerate when the sublevel set is compact, either in terms of vertices for LPs or all points for 0-1 problems. Such enumeration forms a core part of our analysis of exemplars. Note that alternative solution generation methods can differ in the order they enumerate or report solutions. We standardize on using an optimal search mode which generates the next closest vertex or point to optimal by objective value for LPs and 0-1 problems, respectively.

3 Benders Decomposition Review

We review several elements of Benders Decomposition in some detail below to establish context for the proofs of our core results. More detailed treatments of Benders Decomposition and its variants are available elsewhere in the literature (e.g. Birge and Louveaux 2011, Conforti et al. 2014). We consider problems of the form:

$$\text{EF}: \qquad \min_{\boldsymbol{x},\boldsymbol{y}} \mathrm{g}(\boldsymbol{x}) + \boldsymbol{q}^T\boldsymbol{y}$$
$$(\boldsymbol{x},\boldsymbol{y}) \in \mathit{\Gamma}.$$

We call this the extensive-form problem (EF), where g is the first-stage value function, $\boldsymbol{q} \in \mathbb{R}^{n_2}$, and $\mathit{\Gamma} := \{(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^{n_1} \times \mathbb{R}_+^{n_2} \mid \boldsymbol{x} \in X,\ \mathcal{W}\boldsymbol{y} + \mathcal{T}\boldsymbol{x} = \boldsymbol{h}\}$. We call $\boldsymbol{x}$ the first-stage variables and $\boldsymbol{y}$ the second-stage variables. We make several assumptions about problem structure. Let $\mathrm{Y}(\boldsymbol{x}) := \{\boldsymbol{y} \in \mathbb{R}_+^{n_2} \mid \boldsymbol{x} \in X,\ \mathcal{W}\boldsymbol{y} + \mathcal{T}\boldsymbol{x} = \boldsymbol{h}\}$. We make the relatively complete recourse assumption: $\forall \bar{\boldsymbol{x}} \in X\, \exists \bar{\boldsymbol{y}} \in \mathrm{Y}(\bar{\boldsymbol{x}})\ s.t.\ \boldsymbol{q}^T\bar{\boldsymbol{y}} < \infty$. We make the dual non-emptiness assumption: $\{\boldsymbol{\pi} \in \mathbb{R}^{n_3} \mid \boldsymbol{\pi}^T\mathcal{W} \leq \boldsymbol{q}\} \neq \varnothing$ where $n_3 = \dim(\boldsymbol{h})$. We make the assumption a minimizer exists: $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathit{\Gamma}$. We make the finite solution assumption: $z^* := \mathrm{g}(\boldsymbol{x}^*) + \boldsymbol{q}^T\boldsymbol{y}^* > -\infty$. The finite solution assumption implies that $\mathrm{g}(\boldsymbol{x}) + \boldsymbol{q}^T\boldsymbol{y}$ is bounded below over $(\boldsymbol{x},\boldsymbol{y}) \in \mathit{\Gamma}$, and can be established by computing z^* or bounding for some $\underline{z} \in \mathbb{R}$ as $z^* \geq \underline{z}$.

We define a projection operator onto the subspace of the first n_1 (or first-stage) variables:

$$\mathrm{proj}_{\mathbb{R}^{n_1}}(\boldsymbol{p}) := \mathcal{M}\boldsymbol{p}, \quad \mathcal{M} \in \mathbb{R}^{n_1 \times \dim(\boldsymbol{p})}, \quad \mathcal{M}_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{1}$$

We then rewrite the extensive-form problem in terms of just the first-stage variables as the projected variable problem (PV):

$$\text{PV}: \qquad \min_{\boldsymbol{x}} \mathrm{g}(\boldsymbol{x}) + \mathrm{Q}(\boldsymbol{x})$$
$$\boldsymbol{x} \in X.$$

The value (or recourse) function Q is defined for $\boldsymbol{x} \in X$:

$$\text{PVF}: \qquad \mathrm{Q}(\boldsymbol{x}) := \min_{\boldsymbol{y}} \ \boldsymbol{q}^T\boldsymbol{y}$$
$$\boldsymbol{y} \in \mathrm{Y}(\boldsymbol{x}).$$

We call this the primal value function (PVF).

We use the strong duality of linear programs to give another way of computing Q:

$$\begin{aligned} \text{DVF}: \qquad \mathrm{Q}(\boldsymbol{x}) = &\max_{\boldsymbol{\pi}} \ \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}) \\ &\boldsymbol{\pi} \in \mathit{\Pi}, \end{aligned} \tag{2}$$

where $\mathit{\Pi} = \{\boldsymbol{\pi} \in \mathbb{R}^{n_3} \mid \boldsymbol{\pi}^T\mathcal{W} \leq \boldsymbol{q}\}$. We call this the dual value function (DVF). Note $\mathit{\Pi}$ is independent of the argument $\boldsymbol{x}$, is fixed, and as noted before, non-empty by assumption. We write Q in terms of the vertices of $\mathit{\Pi}$ by leveraging the assumptions made earlier. For $\boldsymbol{x} \in X$, $\mathrm{Q}(\boldsymbol{x})$ takes one of three possible states: unbounded above, unbounded below, and finite. The relatively complete recourse assumption rules out unbounded above. The finite solution assumption rules out unbounded below as $\mathrm{g}(\boldsymbol{x}) + \boldsymbol{q}^T\boldsymbol{y} > -\infty, \forall (\boldsymbol{x},\boldsymbol{y}) \in \mathit{\Gamma}$ implies $\mathrm{Q}(\boldsymbol{x}) > -\infty$. We are left with the finite case that requires $\mathit{\Pi}$ to be non-empty, since duality requires $\forall \boldsymbol{x} \in X,\ \exists \bar{\boldsymbol{\pi}} \in \mathit{\Pi}\ s.t.\ \mathrm{Q}(\boldsymbol{x}) = \bar{\boldsymbol{\pi}}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x})$. We use the Fundamental Theorem of Linear Programming (Bertsekas et al. 2013, Prop 3.4.2) to write Q in terms of the dual vertices:

$$\mathrm{Q}(\boldsymbol{x}) = \max_{\boldsymbol{\pi} \in V(\mathit{\Pi})} \ \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}),$$

where $V(S)$ represents the list of vertices of set S. This allows us to write an equivalent problem to the projected variable problem:

$$\begin{aligned} \text{EV}: \qquad &\min_{\boldsymbol{x},\theta} \ \mathrm{g}(\boldsymbol{x}) + \theta \\ &\theta \geq \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}) \quad \forall \boldsymbol{\pi} \in V(\mathit{\Pi}) \\ &\boldsymbol{x} \in X \subseteq \mathbb{R}^{n_1}, \theta \in \mathbb{R}. \end{aligned}$$

We call this the epigraphical variant (EV) problem. For an optimal solution $\boldsymbol{p}^{(2)} = (\bar{\boldsymbol{x}}^{(2)}, \theta^{(2)})$ to the epigraphical variant problem, the projection $\text{proj}_{\mathbb{R}^{n_1}}(\boldsymbol{p}^{(2)}) = \bar{\boldsymbol{x}}^{(2)}$ is an optimal solution to the projected variable problem. The converse is true: for and optimal solution $\bar{\boldsymbol{x}}^{(3)}$ to the projected variable, $(\bar{\boldsymbol{x}}^{(3)}, \bar{\boldsymbol{y}}^{(3)})$ is an optimal solution to the epigraphical variant problem, where $\bar{\boldsymbol{y}}$ exists by the relatively complete recourse assumption and $\boldsymbol{q}^T \bar{\boldsymbol{y}}^{(3)} = \mathrm{Q}(\bar{\boldsymbol{x}}^{(3)})$.

Next we approximate Q by considering only a subset of the dual vertices, $\hat{V} \subseteq V(\Pi)$:

$$\mathrm{Q}_{\hat{V}}(x) := \max_{\boldsymbol{\pi} \in \hat{V}} \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}).$$

Since the maximum function is monotonically increasing, we know that:

$$\mathrm{Q}_{V(\Pi)}(\boldsymbol{x}) = \mathrm{Q}(\boldsymbol{x}) \geq \mathrm{Q}_{\hat{V}}(\boldsymbol{x}) \quad \forall \ \hat{V} \subseteq V(\Pi), \forall \boldsymbol{x} \in X. \tag{3}$$

We now define a version of the epigraphical variant problem relying on $\hat{V}$ instead of $V(\Pi)$, where $\hat{V} \subseteq V(\Pi)$:

$$BM(\hat{V}): \qquad \min_{\boldsymbol{x}, \theta} \ \mathrm{g}(\boldsymbol{x}) + \theta \tag{4a}$$

$$\theta \geq \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}) \quad \forall \boldsymbol{\pi} \in \hat{V} \tag{4b}$$

$$\boldsymbol{x} \in X \subseteq \mathbb{R}^{n_1}, \theta \in \mathbb{R}. \tag{4c}$$

We call this the Benders master problem for $\hat{V}$ ($BM(\hat{V})$). By construction, the Benders master problem gives the same optimal first-stage solution(s) and objective value when $\hat{V} = V(\Pi)$. We include a basic version of the single-cut Benders Algorithm as Algorithm 1.

Algorithm 1 Benders Decomposition Algorithm

```
procedure BENDERS(tol, iterLimit, BM, Q)
    V̂ ← ∅
    while |V̂| ≤ iterLimit do
        Solve(BM(V̂)), x̄ ← x*, θ ← θ*, z ← g(x*) + θ*
        Solve(Q(x̄)), π ← π*, θ̂ ← Q(x̄)
        if ‖θ − θ̂‖ > tol then
            V̂ = V̂_{t−1} ∪ {π}
        else
            break
        end if
    end while
    return (x̄, θ, z, V̂)
end procedure
```

4 Extending Benders to Yield Alternative Solutions

While the ability to automatically generate alternative solutions for LPs and MIPs directly from solvers or modeling languages is useful, this capability does not naturally exist for problems solved by Benders Decomposition. When we use a problem decomposition, we also split up and approximate core structural information that enabled previous tools to do automatic generation of alternative solutions (Section 4.1). To address this, we present methods for treating each of the first-stage and the second-stage alternative solutions (Sections 4.2 and 4.3 respectively), resulting in a combined process capable of treating first-stage, second-stage, and extensive-form alternative solutions. The overall process becomes: prove a problem meets (or modify to enforce) the assumptions from Section 3, divide the problem into first-stage and second-stage components to form the Benders master problem and subproblem, apply Algorithm 2 for tolerances of interest to get first-stage alternative solutions, and then (if needed) apply the LP alternative solution techniques of Section 4.3. We then analyze the resulting alternative solutions for insights into our optimization problems as seen on examples in Sections 5 and 6.

4.1 Impact of Approximation in Benders Decomposition

We know choice of $\hat{V} \subseteq V(\Pi)$ can change the objective value and minimizers in the Benders master problem. As a result, we need to consider the impact this approximation has on the EX-ALT and A-ALT sets, which we do by comparing the sublevel sets. We define the following sublevel sets for the Benders master problem and the extensive-form problems:

$$\mathrm{S}_{BM(\hat{V})}(\tau) = \mathrm{S}(\mathrm{G}_\theta, \mathrm{epi}(\mathrm{Q}_{\hat{V}}), \tau) \tag{5}$$

$$\mathrm{S}_{EF}(\tau) = \mathrm{S}(\mathrm{G}_q, \Gamma, \tau), \tag{6}$$

where $\mathrm{G}_\theta(\boldsymbol{x}, \theta) = \mathrm{g}(\boldsymbol{x}) + \theta$, $\mathrm{G}_q(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{g}(\boldsymbol{x}) + \boldsymbol{q}^T\boldsymbol{y}$, and $\mathrm{epi}(\mathrm{Q}_{\hat{V}})$ is defined relative to epi(Q) as:

$$\mathrm{epi}(\mathrm{Q}) = \{(\boldsymbol{x}, \theta) \in X \times \mathbb{R} \mid \theta \geq \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}), \forall \boldsymbol{\pi} \in V(\Pi)\} \tag{7}$$

$$\mathrm{epi}(\mathrm{Q}_{\hat{V}}) = \{(\boldsymbol{x}, \theta) \in X \times \mathbb{R} \mid \theta \geq \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\boldsymbol{x}), \forall \boldsymbol{\pi} \in \hat{V}\} \tag{8}$$

The following theorem gives us the guarantee that any first-stage alternative solutions for the extensive-form problem will be alternative solutions in the Benders master problem:

Theorem 4.1. *$proj_{\mathbb{R}^{n_1}}(S_{EF}(\tau)) \subseteq proj_{\mathbb{R}^{n_1}}(S_{BM(\hat{V})}(\tau)),\ \forall \hat{V} \subseteq V(\Pi), \forall \tau \in \mathbb{R}$.*

Proof. Proof This follows directly by the nature of the projection operator of (1) and Lemmas 3 and 4 from the Proof Appendix. □

Since this result is defined over the first-stage variables, it is abstracting away the details of the second-stage variables for the extensive-form problem. If $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau)) \neq \varnothing$ for some fixed $\tau \in \mathbb{R}$, we know there are feasible second-stage points in the sublevel set (e.g. $\bar{\boldsymbol{x}} \in \mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau)) \rightarrow |(\{\bar{\boldsymbol{x}}\} \times \mathrm{Y}(\bar{\boldsymbol{x}})) \cap \mathrm{S}_{EF}(\tau)| \geq 1$).

The core insight for the sublevel set properties comes from analyzing epi(Q) and $\mathrm{epi}(\mathrm{Q}_{\hat{V}})$, which gives the following result:

Theorem 4.2. *$epi(Q) \subseteq epi(Q_{\hat{V}}), \forall\ \hat{V} \subseteq V(\Pi)$*

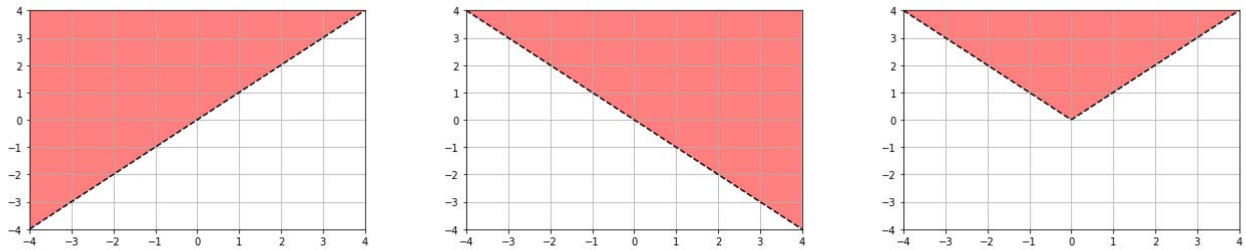
Proof. Proof The inclusion holds trivially if $\mathrm{epi}(\mathrm{Q}) = \varnothing$. In the case $\mathrm{epi}(\mathrm{Q}) \neq \varnothing$, we consider $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q})$. We have $(\bar{\boldsymbol{x}}, \bar{\theta}) \in X \times \mathbb{R}$ by definition of epi(Q). All that remains is showing that $\bar{\theta} \geq \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\bar{\boldsymbol{x}}), \forall \boldsymbol{\pi} \in \hat{V} \subseteq V(\Pi)$ holds and this follows directly from $\bar{\theta} \geq \boldsymbol{\pi}^T(\boldsymbol{h} - \mathcal{T}\bar{\boldsymbol{x}}), \forall \boldsymbol{\pi} \in V(\Pi)$ as part of the definition of epi(Q). So $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q}_{\hat{V}})$ giving inclusion in the non-empty case. □

We do know that the converse results are not true in general:

Remark 4.1. *$proj_{\mathbb{R}^{n_1}}(S_{BM(\hat{V})}(\tau)) \subseteq proj_{\mathbb{R}^{n_1}}(S_{EF}(\tau))$ does not hold in general.*

Remark 4.2. *$epi(Q_{\hat{V}}) \subseteq epi(Q)$ does not hold in general.*

The core intuition for all of the results can be seen in the following example where $\mathrm{Q}(\boldsymbol{x}) = |\boldsymbol{x}|$:



(a) $\text{epi}(Q_{\hat{V}_1}),\ Q_{\hat{V}_1}(x) = x$ (b) $\text{epi}(Q_{\hat{V}_2}),\ Q_{\hat{V}_2}(x) = -x$ (c) $\text{epi}(Q_{\hat{V}_1}),\ Q_{\hat{V}_1}(x) = |x|$

Figure 1: Examples of $\text{epi}(Q_{\hat{V}})$ for several values of $\hat{V}$. $\hat{V}_1 = \{1\},\ \hat{V}_2 = \{-1\},\ \hat{V}_3 = \{-1, 1\}$

We see the epigraph containment result of Theorem 4.2: the first two plots contain the third. The converse of Theorem 4.2 does not hold for this example. The figure also gives us an intuition about Theorem 4.1 for two reasons. First, the main difference between the the extensive-form and Benders master problems is the difference between the exact and approximate value functions. Second, we see the geometric connection between the epigraph and sublevel sets of a function.

4.2 Extracting Alternative Solutions for First-Stage Variables

The extensive-form and Benders master problems share the first-stage variables. As a consequence of this, we break the generation of alternative solutions to the extensive-form problem into two steps. First, we find alternative solutions over the first-stage variables. Second, given a solution for the first-stage variables we generate alternative solutions for the second-stage variables, which we defer to Section 4.3. We note that there are cases where all we need are first-stage decisions (e.g. two-stage stochastic programming for Capacity Expansion); in such cases, being able to generate first-stage decisions separately is a major benefit of AOS-Benders method. We present a three-step method in Algorithm 2 to achieve this first-stage alternative solution generation.

Algorithm 2 AOS-Benders Algorithm

procedure BENDERS($BendersTol$, $OptTol$, $iterLimit$, $solLimit$, BM, Q, g, AOSKernel)
  $(\boldsymbol{x}^*, \theta, z^*, \hat{V}) \leftarrow$ Benders($BendersTol$, $iterLimit$, BM, Q) ▷ Step 1: *Benders Solve*
  $\tau \leftarrow z^* + OptTol$
  $S_{BM} \leftarrow$ AOSKernel($BM(\hat{V}), solLimit, \tau$) ▷ Step 2: *Generate Candidates*
  $S_{proj(EF)} \leftarrow \varnothing$
  for $(\bar{\boldsymbol{x}}, \bar{\theta}) \in S_{BM}$ **do** ▷ Step 3: *Certify Candidates*
    Solve($Q(\bar{\boldsymbol{x}})$), $\theta \leftarrow Q(\bar{\boldsymbol{x}})$
    if $g(\bar{\boldsymbol{x}}) + \theta \leq \tau$ **then**
      $S_{proj(EF)} \leftarrow S_{proj(EF)} \cup \{\bar{\boldsymbol{x}}\}$
    end if
  end for
  return $S_{proj(EF)}$
end procedure

There are three steps in AOS-Benders. The first step is *Benders Solve*, which solves the Benders master problem to optimality and returns the optimal objective value, z^*, and terminating cuts, $\hat{V}$. For example, we can apply Algorithm 1. In step two, *Generate Candidates*, we generate alternative solutions to $S_{BM(\hat{V})}(\tau)$. We do this by calling a function, $AOSKernel$, to generate alternative solutions for the Benders master problem for $\hat{V}$. For example, $AOSKernel$ can be one of the methods discussed in Section 2.2 that is suitable for the Benders master problem. The final step is to *Certify Candidates*, which filters the points generated in

Step 2 to keep only solutions that are in $\text{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$. For each $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{S}_{BM}$, we test if $\mathrm{g}(\bar{\boldsymbol{x}}) + \mathrm{Q}(\bar{\boldsymbol{x}}) \leq \tau$, which is sufficient to guarantee that $\bar{\boldsymbol{x}} \in \text{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$ from the definition (6).

4.3 Extracting Alternative Solutions for Second-Stage Variables

Alternative solutions for second-stage variables rely on having the first-stage variables fixed to a specific value. We assume that we have some fixed $\bar{\boldsymbol{x}} \in X$. We discussed in Section 4.1 how the approximation in the Benders master problem impacts generating alternative solutions to the extensive-form problem, but this impact was localized to the first-stage variables and therefore the projected variable problem. As a result, nothing in the value function Q an approximation, which means that the generation of alternative solutions for the second-stage variables can ignore the details of the Benders master problem.

Here we are concerned with taking a previously-known point, $\bar{\boldsymbol{x}} \in X$, from the projected variable problem and generating alternative solutions for the second-stage variables. This will also let us generate alternative solutions to the overall extensive-form problem. The value function, Q, is an LP so we can apply any of the alternative solutions generation methods for LPs described in Section 2.2 like Pyomo-AOS-Linear.

To generate alternative solutions to the extensive-form problem, we need to address the difference between the $\text{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$ and $\mathrm{S}_{EF}(\tau)$ sublevel sets. We recall the definitions as:

$$\mathrm{S}_{EF}(\tau) = \mathrm{S}(\mathrm{G}_q, \mathit{\Gamma}, \tau) \subseteq \mathbb{R}^{n_1+n_2}.$$

Note that the definition of $\mathrm{S}_{EF}(\tau)$ makes no use of the fact that we have a fixed $\bar{\boldsymbol{x}} \in X$. We use this to define the sublevel set of the remaining $\boldsymbol{y}$ options for a fixed $\bar{\boldsymbol{x}} \in X$:

$$\mathrm{S}^P(w; \bar{\boldsymbol{x}}) = \mathrm{S}(\boldsymbol{q}^T\boldsymbol{y}, \mathrm{Y}(\bar{\boldsymbol{x}}), w).$$

We also then need to take into account the combined optimality tolerance for extensive-form alternative solutions by setting the second-stage optimality tolerance in terms of both the overall tolerance, τ, and the impact of the first-stage decision, $\bar{\boldsymbol{x}} \in X$. We see how this works in the following theorem to reconstruct combined alternative solutions for the extensive-form:

Theorem 4.3. *Suppose τ given and $\bar{\boldsymbol{x}} \in \text{proj}_{\mathbb{R}^{n_1}}(S_{EF}(\tau))$. Let $\hat{\boldsymbol{y}} \in S^P(\tau - g(\bar{\boldsymbol{x}})); \bar{\boldsymbol{x}})$, then $(\bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in S_{EF}(\tau)$.*

Proof. Proof Show that $(\bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in \mathit{\Gamma}$ and $\mathrm{g}(\bar{\boldsymbol{x}}) + q^T\hat{\boldsymbol{y}} \leq \tau$ to recover the definition of $\mathrm{S}_{EF}(\tau)$.
$\bar{\boldsymbol{x}} \in \text{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$ gives $\bar{\boldsymbol{x}} \in X$ by definition. Since $\hat{\boldsymbol{y}} \in \mathrm{S}^P(\tau - \mathrm{g}(\bar{\boldsymbol{x}}))$ gives $\hat{\boldsymbol{y}} \in Y(\bar{\boldsymbol{x}})$, then holds $(\bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in \mathit{\Gamma}$ by construction. Since $\hat{\boldsymbol{y}} \in \mathrm{S}^P(\tau - g(\bar{\boldsymbol{x}}); \bar{\boldsymbol{x}})$, we know by construction $q^T\hat{\boldsymbol{y}} \leq \tau - g(\bar{\boldsymbol{x}})$. Rearranging gets $\mathrm{g}(\bar{\boldsymbol{x}}) + q^T\hat{\boldsymbol{y}} \leq \tau$. □

This makes generating extensive-form alternative solutions for fixed first-stage decisions, $\bar{\boldsymbol{x}} \in X$, a matter of generating linear programming alternative solutions with the second-stage optimality tolerance $w = \tau - \mathrm{g}(\bar{\boldsymbol{x}})$. As a result, generating extensive-form alternative solutions is an optional post-processing step to making first-stage alternative solutions with Algorithm 2.

5 Application: Farmer's Problem

We now illustrate how AOS-Benders generates alternative solutions for stochastic programming problems. We consider a farmer planning problem from Birge and Louveaux (2011) where the farmer has a set of 3 crops (wheat, corn, and sugar beets) and cattle to feed. In the first-stage, the farmer controls how many acres of land of each crop are planted. In the second-stage, the farmer either buys or sells crops to meet a cattle feed target and to minimize financial loss.

5.1 Model

The scenario-based stochastic version with variable crop yields is:

$$
\begin{aligned}
&\min_{\boldsymbol{x}}(150x_1 + 230x_2 + 260x_3) \\
&\qquad + \sum_{\omega \in \Omega} p^{(\omega)} \left[(238y_1^{(\omega)} + 210y_2^{(\omega)}) - (170w_1^{(\omega)} + 150w_2^{(\omega)} + 36w_3^{(\omega)} + 10w_4^{(\omega)}) \right] \\
&x_1 + x_2 + x_3 \leq 500 \\
&\xi_1^{(\omega)} x_1 + y_1^{(\omega)} - w_1^{(\omega)} \geq 200 \; \forall \, \omega \in \Omega \\
&\xi_2^{(\omega)} x_2 + y_2^{(\omega)} - w_2^{(\omega)} \geq 240 \; \forall \, \omega \in \Omega \\
&\xi_3^{(\omega)} x_3 - w_3^{(\omega)} - w_4^{(\omega)} \geq 0 \; \forall \, \omega \in \Omega \\
&w_3^{(\omega)} \leq 6000 \\
&\boldsymbol{x}, \boldsymbol{y}^{(\omega)}, \boldsymbol{w}^{(\omega)} \geq \boldsymbol{0} \; \forall \, \omega \in \Omega
\end{aligned}
$$

Here $\boldsymbol{\xi}^{(\omega)}$ is a vector denoting the yields-per-acre planted under scenario ω. There are three crops: wheat (1), corn (2), and beets (3 & 4). $\boldsymbol{x}$ is the number of acres of each crop planted. $\boldsymbol{y}$ and $\boldsymbol{w}$ are the number of tons of crops purchased and sold respectively. $\boldsymbol{w}$ has two values for sugar beets to reflect the two sale points. The following equations are used to decompose this model into the form used by Benders Decomposition:

$$
\begin{aligned}
\mathrm{g}(\boldsymbol{x}) &= 150x_1 + 230x_2 + 260x_3 \\
X &= \{\boldsymbol{x} \in \mathbb{R}_+^3 \;\mid\; x_1 + x_2 + x_3 \leq 500\} \\
\mathrm{Q}(\boldsymbol{x}) &= \sum_{\omega \in \Omega} p^{(\omega)} \mathrm{Q}_k(\boldsymbol{x}, \boldsymbol{\xi}^{(\omega)})
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Q}_k(\boldsymbol{x}, \boldsymbol{\xi}) = \min_{\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{u}} &(238y_1 + 210y_2) - (170w_1 + 150w_2 + 36w_3 + 10w_4) \\
&\xi_1 x_1 + y_1 - w_1 \geq 200 \\
&\xi_2 x_2 + y_2 - w_2 \geq 240 \\
&\xi_3 x_3 - w_3 - w_4 \geq 0 \\
&w_3 \leq 6000 \\
&\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{u} \geq \boldsymbol{0}
\end{aligned}
$$

Again, slack variables $\boldsymbol{u}$ can be introduced to convert from inequality to equality constraints, but we omit them to simplify our presentation.

Both the single-scenario and multiple-scenario cases of this problem are linear programs with continuous variables. This means there are convex sublevel sets in both the extensive-form and Benders master problems. We also have relatively complete recourse in the value function, because we can always choose $\bar{\boldsymbol{y}} = [200 - \xi_1 x_1,\ 240 - \xi_2 x_2]^T$ and $\boldsymbol{w} = 0$ for a cost of $[238,\ 210]\bar{\boldsymbol{y}}$ for $\boldsymbol{x} \in X$. We note that we have at least one dual vertex because $\mathrm{Q}(\boldsymbol{0}, \boldsymbol{\xi}) = [238,\ 210][200,\ 240]^T = 98000$ for all $\boldsymbol{\xi} \in \mathbb{R}^3$. We have z^* bounded below because $\mathrm{Q}(\boldsymbol{x}, \boldsymbol{\xi}) \geq -500 \max\{170\xi_1, 150\xi_2, 36\xi_3\}$, which corresponds to growing only the most profitable crop and selling it all to the market. So long as $\boldsymbol{\xi} \in \mathbb{R}^3$, we have $z^* \in \mathbb{R}$ satisfying the finite solution assumption. This suffices to demonstrate that both the deterministic and stochastic Farmer's problem fit into our framework and assumptions.

5.2 Meaning of Alternative Solutions

There are two key things to note about the meaning of these alternative solutions. First, we can generate alternative solutions in the X (or projected variable) space. As this is a staged stochastic problem, this means that we can generate alternative solutions for first-stage variables without generating corresponding second-stage variables. This significantly reduces the complexity of generating alternative solutions for first- and second-stage variables. Even in this simple problem with N scenarios, this reduces the size of the variable space to explore first-stage alternative solutions from $\mathbb{R}_+^{3+5N}$ to $\mathbb{R}_+^3$.

Second this is a continuous problem, so we may have an infinite number of points in any A-ALT set even if the EX-ALT set has a single point. This means that we need to consider techniques that emphasize discovery of the "interesting" or "meaningful" alternative solutions. What "interesting" means can be problem-specific. In some cases, a structured vertex representation may suffice as with the Pyomo-AOS-Linear method we use here. In others, solution diversification strategies may be needed (e.g. Danna et al. 2007, Petit and Trapp 2019).

5.3 Extracting Alternative Solutions from the Farmer's Problem

5.3.1 Single-Scenario Problem

We tested AOS-Benders on the single-scenario mean-yield Farmer's Problem by choosing $|\Omega| = 1$ and $\boldsymbol{\xi} = [2.5, 3, 20]^T$. After 9 iterations and 8 cuts, the Benders Algorithm converges to the known optimal point $z^* = -118,600$, $\boldsymbol{x}^* = [120, 80, 300]^T$, $\boldsymbol{y}^* = [0, 0]^T$, $\boldsymbol{w}^* = [100, 0, 6000]$. We use the Pyomo-AOS-Linear method from Pyomo 6.8.1 to generate alternative solutions for both the Benders master problem (i.e. the Algorithm 2 AOSKernel) and subproblem. This method enumerates up to K vertices subject to an optimality tolerance according to a search mode. We use the 'optimal' search mode, which orders vertices on the basis of the original objective, and we use an absolute optimality tolerance relative to the optimal objective value. In the limit, this reduces to exhaustive discovery of the vertices of the constraint space. As a result, when less than K solutions are returned, we have exhaustively enumerated the feasible vertices.

First, we enumerated exact optimal solutions (0% optimality tolerance) and found that there was only one point, the optimal solution to the Benders master problem for these cuts. By Theorem 4.1, we know that all the exact optimal first-stage solutions to the extensive-form problem are contained in this set. So we have found the one exact optimal first-stage candidate solution to the extensive-form problem and have it certified as a true solution by the termination condition for Algorithm 1. Next we enumerate for the subproblem given $\boldsymbol{x}^*$: we find only one solution, which is the known recourse decision as an exact optimal solution. This matches expectations: the farmer is likely to take a unique cost-optimal recourse.

Second, we enumerated approximate optimal solutions given a 1% optimality tolerance; given $z^* = -118,600$, this means means we consider solutions with objective values in $[-118,600, -117,414]$ (or $\mathrm{S}_{BM(\hat{\mathcal{V}})}(\tau)$, $\tau = -117,414$). We set $K = 10$ and generated 6 points ($\boldsymbol{x}^*$ and 5 new points), which are the vertices of $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{\mathcal{V}})}(\tau))$. All 5 new solutions had objectives of $-117,414$ indicating that they are exactly along the 1% optimality boundary. All 5 of these new vertices in $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{\mathcal{V}})}(\tau))$ then passed the certification step in Algorithm 2. We then know by Theorem 4.1 and compactness of X that we have $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(-117,414)) = \mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{\mathcal{V}})}(-117,414))$. This means we have, and can prove we have, a representation of the 1% optimal first-stage decisions as the convex combination of these 6 points.

5.3.2 Multiple-Scenario Problem

We tested our generation of alternative solutions on the stochastic farmer's problem with three scenarios. The first scenario was the same as the mean yield used in the single-scenario version (i.e. $\boldsymbol{\xi} = [2.5, 3, 20]^T$). The second and third scenarios were yields of 20% above and below mean across crops. It took 11 iterations and 10 cuts for Benders Decomposition to converge. The expected optimal solution of $\boldsymbol{x}^* = [170, 80, 250]$ with $z^* = -108,390$ was discovered. We use the same enumeration methods as in the deterministic case with the same exact and then approximate solution exploration approach. When we use Pyomo-AOS-Linear with optimality tolerance 0%, we get only one point: $\boldsymbol{x}^*$. By Theorem 4.1, we conclude $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(z^*)) = \{\boldsymbol{x}^*\}$, meaning we have found the only exact optimal first-stage decision. We then generated alternative solutions for second-stage variables given $\boldsymbol{x}^*$, and we again get only one solution per scenario.

Next, we consider the generation of approximate solutions with two optimality tolerances: 1% and 50%. When the optimality tolerance is 1%, $\tau = -107,306.1$. We get 15 possible solutions from Pyomo-AOS-

Linear when $K = 50$; these are the 15 vertices for $\mathrm{S}_{BM(\hat{V})}(-107, 306.1)$. After we apply the certification step of Algorithm 2, only 11 points are in $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(-107, 306.1))$ for the first-stage decisions. In the 50% optimality case ($\tau = -54, 195$) again with $K = 50$, we get 43 vertices for $\mathrm{S}_{BM(\hat{V})}(-54, 195)$. After the certification step, only 29 are in $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(-54, 195))$ for the first-stage decisions. In both 1% and 50% optimality tolerance, we discover $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{V})}(\tau)) \neq \mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$ since not all vertices from $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{V})}(\tau))$ are admitted to $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$.

5.4 Discussion

This example illustrates that we can generate alternative solutions for stochastic problems solved by Benders Decomposition. We are able to prove the uniqueness of the optimal crop planting strategy even with the problem decomposed in both the deterministic and stochastic cases. We also see that in both the one- and three-scenario cases the majority of the A-ALT Benders master solutions are also first-stage solutions to the corresponding A-ALT extensive-form problem. This shows that even approximate alternative solutions discovered using Benders Decomposition can map to alternative solutions to the first-stage extensive-form problem. This is likely a function of having sufficient density of cuts near the optimal solutions. Finally, we are able to determine when $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{V})}(\tau)) = \mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EF}(\tau))$ holds by exhausting the vertices of $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{BM(\hat{V})}(\tau))$.

6 Application: Israeli-Wood Shortest Path Interdiction

The following example illustrates the use of AOS-Benders on an interdiction problem described by Israeli and Wood (2002). They present a variant on the shortest path interdiction problem described as "maximize the shortest $s - t$ path length in a directed network by interdicting arcs" or the Maximizing the Shortest Path (MXSP) problem. They treated this problem with both Benders Decomposition and several advanced Benders Decomposition variants. MXSP models two agents, a defender, and an attacker. The defender (or "network user") wants to cross from s to t. The attacker (or "interdictor") wants to make that as high a cost as possible. In this version of the problem, the attacker is worsening the best-case $s - t$ traversal in the network, and the objective value is the cost of that best traversal.

Knowing whether multiple traversals exist that achieve the optimal value has a clear value in the security context, and identifying what (or some of) those optimal traversals are also has clear value in the security context. We can also move beyond the optimal traversal to consider near optimal traversals. This is especially useful when the instance data is an estimate or noisy. Asking the same two questions about near-optimal traversals has clear value if this attacker-defender problem is used to plan defender reaction. Similarly, the same questions about optimal traversal existence and identification apply to optimal (and near-optimal) attacks as well.

6.1 Models

We adapt the Extensive Form and Bender Decomposition models that Israeli-Wood generated for this problem (specifically their Algorithm 1 and 1-E models).

6.1.1 Original Formulation

The MXSP interdiction problem is a max-min problem where the attacker decides how to interdict paths for the defender. The defender minimizes its cost to traverse from s to t on graph $G = (N, A)$. The defender traversal route is recorded by y_k where $y_k = 1$ if traversed and 0 otherwise. This means that a defender path is defined as $\{k \in A \mid y_k = 1\}$. The attacker interdiction choices are recorded by x_k where $x_k = 1$ if interdicted and 0 otherwise. The source node is s and the sink node is t. We use special sets to denote arcs entering and exiting node i as $\mathrm{RS}(i)$ and $\mathrm{FS}(i)$ respectively. Arcs have two components to their traversal cost, the baseline cost $c_k \in [0, \infty)$ and the added cost if interdicted $d_k \in [1, \infty)$. The attacker has a budget of $m \in [1, \infty)$ available to spend, and each arc has an interdiction cost $r_k \in [1, \infty)$. Note that we do not explicitly enforce the integrality of $\boldsymbol{y}$. The constraint structure of the inner minimization problem is totally unimodular, which induces integrality (see Conforti et al. 2014, Section 4.2). We assume that our solver returns a vertex solution (e.g. like the simplex method).

The max-min model fits into our extensive-form structure in Section 3 by applying the transform $\max_{a \in A} \mathrm{f}(a) = \min_{a \in A} -\mathrm{f}(a)$. Thus we have:

$$
\begin{aligned}
&\min_{\boldsymbol{x} \in X} \min_{\boldsymbol{y}} - \sum_{k \in A} (c_k + x_k d_k) y_k \\
&\sum_{k \in \mathrm{FS}(i)} y_k - \sum_{k \in \mathrm{RS}(i)} y_k = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \forall i \in N \backslash \{s, t\} \end{cases} \\
&y_k \geq 0 \quad \forall k \in A \\
&X = \{\boldsymbol{x} \in \{0,1\}^{|A|} \quad | \; \boldsymbol{r}^T \boldsymbol{x} \leq m\}
\end{aligned} \tag{9}
$$

The first-stage elements of Benders Decomposition are:

$$
\mathrm{g}(\boldsymbol{x}) := 0, \;\; X := \{\boldsymbol{x} \in \{0,1\}^{|A|} \quad | \; \boldsymbol{r}^T \boldsymbol{x} \leq m\}.
$$

The second-stage elements are:

$$
\mathrm{Q}(\boldsymbol{x}) = \min_{y} - \sum_{k \in A} (c_k + x_k d_k) y_k \tag{10a}
$$

$$
s.t. \qquad \sum_{k \in \mathrm{FS}(i)} y_k - \sum_{k \in \mathrm{RS}(i)} y_k = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \forall i \in N \backslash \{s, t\} \end{cases} \tag{10b}
$$

$$
y_k \geq 0 \quad \forall k \in A \tag{10c}
$$

This interdiction problem takes on special meaning; the Q is the value function of the defender's response to an attack.

6.1.2 Benders Decomposition Form

The resulting model fits into Section 3 format:

$$
\min_{\boldsymbol{x} \in X, u \in \mathbb{R}} \quad u \tag{11a}
$$

$$
u \geq \; -\boldsymbol{c}^T \boldsymbol{y} - \boldsymbol{x}^T \mathcal{D} \boldsymbol{y} \quad \forall \hat{\boldsymbol{y}} \in \hat{V} \tag{11b}
$$

Here $\mathcal{D} = diag(\boldsymbol{d})$. The constraint structure in (11b) matches the structure of the objective in (10a). This leads to an interpretation of Q in (10) as the dual definition of the value function, (2). Both of these features are interdiction modeling structure addressed in Brown et al. (2006).

The overall approach to solving the MXSP problem is to generate select candidate paths through the network via the subproblems, and only needing to optimize against those in the master problem. It is the selective generation of candidate paths that motivates using Benders Decomposition for this problem. We only consider the "short" paths for the attacker to interdict in the master problem and the delayed constraint generation approach allows us to only consider those paths "short" enough to be of interest to the attacker. We compare this to the master problem that Israeli-Wood define as:

$$
\begin{aligned}
\text{Master}(\hat{V})\text{-1a:} \qquad & \max_{\boldsymbol{x} \in X, z \in \mathbb{R}} z \\
& z \leq \boldsymbol{c}^T \hat{\boldsymbol{y}} + \boldsymbol{x}^T \mathcal{D} \hat{\boldsymbol{y}} \quad \forall \hat{\boldsymbol{y}} \in \hat{Y}
\end{aligned} \tag{12}
$$

We see that the difference between the two master problems is $u^* = -z^*$. Note that Israeli-Wood call their dual vertices object $\hat{Y}$. In this problem, each dual vertex, $\hat{\boldsymbol{y}}$, has meaning as a specific defender $s - t$ path, making the collection of paths $\hat{Y}$. We maintain our format for standard dual vertices, $\hat{V}$ and $V(\Pi)$, since it is more general.

6.1.3 Analyzing the Israeli-Wood Model

There are several structural components of this model that relate to our Benders Decomposition methodology described in Section 3. The Benders Form of Israeli-Wood relies on a specific part of the max-min structure that lets the subproblem of (10) serve as the DVF form of the value function. This enables the cuts in (11) to be defined over the same variables as in the extensive-form. This is a more general principle of interdiction modeling and relates to the difference between "capacity" and "cost" interdiction modeling discussed in Brown et al. (2006). The subproblem has an LP structure, and it is also clear that optimal vertices are integer valued as a result of the total unimodularity of the network flow constraint structure. Additionally, the subproblem is the well known shortest s-t path problem, and Israeli and Wood leverage specialized algorithms to find shortest paths (i.e. Byers and Waterman 1984).

The dual non-emptiness assumption corresponds to the existence of a feasible $s - t$ flow. We satisfy relatively complete recourse and finiteness assumptions by noting the optimal cost is bounded above by the fully interdicted s-t path and bounded below by Q(0). Thus, our assumptions are satisfied if A is finite and $c_k, d_k \in \mathbb{R}_+, \forall k \in A$. The original problem is not convex because the $\boldsymbol{x}$ are integral, so the master problem remains nonconvex. Benders Decomposition splits the MILP structure into a binary program and a linear program. This has advantages for the generation of alternative solutions, we can avoid limitations of commercial solvers when generating alternative solutions (see Section 2.2).

6.2 Extracting Alternative Solutions From Israeli-Wood

We demonstrate our methods on the following MXSP problem. We exhaustively generate possible optimal actions of the attacker and defender. This allows us to examine the overlap in optimal actions for both agents across the number of attacked arcs, which goes beyond a traditional sensitivity analysis. Our examples rely on the following directed graph where we flow from s to t:

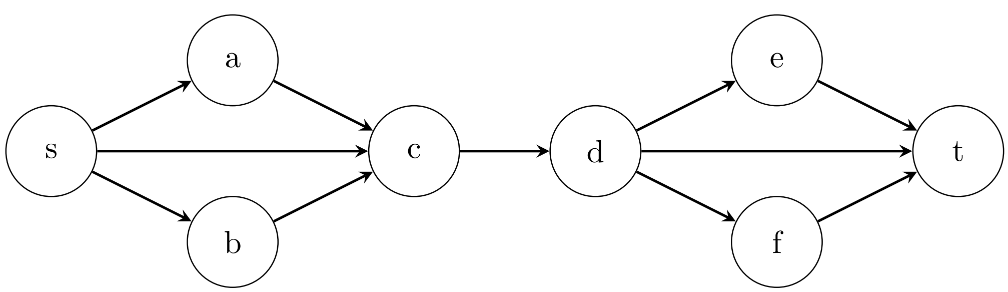


Figure 2: Network used for s-t Flow Interdiction.

We set $c_k = 1$ and $d_k = 3$ for all arcs $k \in A$. All arcs have the same default cost to transit, and interdicted arcs are still passable but cost enough to force avoidance if possible. We allow the attacker to choose m arcs, so $X = \{\boldsymbol{x} \in \{0,1\}^{|A|} \;\mid\; \sum_{k \in A} x_k \leq m\}$. We apply our AOS-Benders process to generate alternative solutions. We use Pyomo-AOS-Binary from Pyomo 6.8.1 to generate alternative solutions for the Benders master problem (i.e. the Algorithm 2 AOSKernel). We use Pyomo-AOS-Linear to generate alternative solutions for the subproblem. In both cases, we use the optimal search mode with the GLPK solver. We report our results in the max-min sense, using non-negative objective values to match the path cost in the MXSP problem.

6.2.1 Attack 1 Arc Example

When the attacker may only attack one arc and the attack impacts are all the same, it makes intuitive sense that the attacker picks arcs that the defender must necessarily traverse to travel from s to t (i.e. chokepoints). In this network, the only chokepoint is $c \to d$. By inspection, we see that this is the unique optimal attack. The attack to interdict $c \to d$ is the only answer that achieves a defender best traversal cost of 6. However, the Benders master problem can converge to the optimal cost of 6 with the interdiction choice as $c \to d$ with different cutpools. The following cuts serve as a certificate of Benders master converging with $\hat{V} \neq V(\Pi)$ and the terminating cutpool, $\hat{V}$, we encountered is:

$$z \leq 3 + 3(x_{s,c} + x_{c,d} + x_{d,t})$$
$$z \leq 4 + 3(x_{s,a} + x_{a,c} + x_{c,d} + x_{d,t})$$

When we run Pyomo-AOS-Binary to exhaustion on Benders master problem: $BM(\hat{V})$, it returns two options with an optimal cost of 6, interdict $c \to d$ or interdict $d \to t$. Since we use the optimal mode with a tolerance of zero, we know that these are the only two possible exact optimal solutions for these cuts by Theorem 4.1. The two options make sense for the cuts in the master problem, but interdicting $d \to t$ needs to be put through the rest of the AOS-Benders' three-step process. In the certification stage, interdicting $d \to t$ has a defender cost of 4, and is therefore suboptimal. As a result, we recover only one exact optimal solution for the attacker:interdicting $c \to d$. We know by Theorem 4.1 this is the only such solution that achieves an optimal cost of 6. We also generated the alternative solutions for the defender to the optimal attack using Pyomo-AOS-Linear. The only exact optimal defender path is $s \to c \to d \to t$.

6.2.2 Attack 2 Arcs Example

When the attacker interdicts 2 arcs, our Benders Algorithm terminates optimally by interdicting $c \to d$ and $s \to c$ with the following cuts, $\hat{V}$:

$$z \leq 3 + 3(x_{s,c} + x_{c,d} + x_{d,t})$$
$$z \leq 4 + 3(x_{s,a} + x_{a,c} + x_{c,d} + x_{d,t})$$
$$z \leq 4 + 3(x_{s,c} + x_{c,d} + x_{d,e} + x_{e,t})$$

Based on the graph symmetry, we might also expect an interdiction strategy of $c \to d$ and $c \to t$ to be optimal. When we run Pyomo-AOS-Binary on this terminating model: $BM(\hat{V})$, we get three options with an optimal cost of 7 according to the Benders master problem:

1. Interdicting $s \to c$ and $c \to d$
2. Interdicting $s \to c$ and $d \to t$
3. Interdicting $c \to d$ and $d \to t$

We can check each of these three solutions with the true subproblem value and get that options 1 and 3 as expected are exact optimal solutions with a true cost of 7. Option 2 is not an exact optimal attack with a true cost of 5. Again by Theorem 4.1, we know that options 1 and 3 are the only exact optimal attacks.

When we look at alternative solutions in the defender response, we get a split in behavior depending on what is attacked. In option 1, the defender can respond by using $s \to a \to c \to d \to t$ or $s \to b \to c \to d \to t$. In option 3, the defender can respond by using $s \to c \to d \to e \to t$ or $s \to c \to d \to f \to t$. The defender paths then only and always overlap in being forced to transit over $c \to d$.

6.2.3 Attack 3 Arcs Example

When the attacker may attack three arcs and the attack impacts are still all the same, we intuitively expect an interdiction strategy like the previous case that cancels out the symmetry by interdicting $s \to c$, $c \to d$, and $d \to t$. This is in fact an optimal solution at which our Benders Algorithm terminates with an optimal defender traversal cost of 8. When we then run Pyomo-AOS-Binary, the only option it returns is the strategy to interdict $s \to c$, $c \to d$, and $d \to t$. We now know that this is the only optimal solution by Theorem 4.1. When we look at the defender response paths, we get alternative solutions there:

1. $s \to a \to c \to d \to e \to t$
2. $s \to b \to c \to d \to e \to t$
3. $s \to a \to c \to d \to f \to t$
4. $s \to b \to c \to d \to f \to t$

The four solutions turn out to be all the valid paths from s to t that avoid both $s \to c$ and $d \to t$.

6.3 Discussion

Our Benders Decomposition approach generates alternative solutions for max-min style interdiction problems. On our specific problem instance, we get a sensitivity analysis that shows several things.

First, $c \to d$ is always interdicted by a competent attacker. Second, the optimal defender response to optimal attack always changes based off attacker strength in this network. The first point is something that could be recognized without alternative solutions by looking for chokepoints. However, the second point leverages the ability to look at all of the exact optimal alternative solutions as attacker strength changes.

7 Conclusion

We demonstrate a generation method and structural theory for alternative solutions for Benders Decomposition. Our AOS-Benders process maintains the core feature Benders Decomposition is known for in trading a single problem over the entire variable space for multiple problems over subsets of variables. The AOS-Benders process can be appended to a traditional Benders Decompositionalgorithm, subject to some technical assumptions, with existing alternative solution generation codes and a certification step. AOS-Benders also provides the capability to still generate alternative solutions when extensive form problems are either intractable or are otherwise undesirable. Such a capability is important on a range of problem classes including large-scale stochastic programming and max-min interdiction problems. We also have a theoretical characterization of the alternative solutions through the sublevel sets. This enables strong claims about first-stage solution properties including exhaustive solution enumeration under variable projection; without AOS-Benders, such claims would either require application-specific theory, require enumeration over full extensive-formsolutions, or be intractable entirely.

The new capabilities provided by our AOS-Benders algorithm raise several new questions about alternative solution generation. First, which technical assumptions made about Benders Decompositioncan be relaxed (e.g. relatively complete recourse or continuous second-stage variables)? Second, can AOS-Benders improve the known scaling challenges (e.g. Lau et al. 2024) of alternative solution generation? Third, while Israeli and Wood (2002) used problem-specific methods to generate multiple subproblem solutions, could recent ML approximation methods for subproblems (e.g. Larsen et al. 2022) help with scaling and what specific cut-pool management rules would best fit the AOS-Benders paradigm? Fourth, there are recent advances in stochastic and bilevel programming that rely on Benders Decomposition paradigms with some adaptation (e.g. Elçi and Hooker 2022, Byeon and Van Hentenryck 2022, respectively), raising which paradigms support AOS-Benders-like alternative solution generation methods? Fifth, which other problem decomposition methods (e.g. Dantzig-Wolfe, Progressive Hedging) admit alternative solution generation methods and under what assumptions?

There are also a range of application areas for AOS-Benders that remain to be explored. The separable generation of alternative solutions has a range of applications from first-stage only generation in long term planning models (e.g. electrical grid capacity expansion) and generation of alternative solutions under privacy concerns. Given the variety of Benders Decomposition modifications and extensions, which of them adapt best to an alternative solutions paradigm (e.g. cut selection rules) remains an open question. All of these questions and applications stand to enhance the new optimization problems and modeling questions that can be treated by alternative solutions generally and under problem decomposition specifically.

Here we treat all the proofs necessary to prove Theorem 4.1 and the converse Remarks from Section 4.1.

A Sublevel Set Technical Results

We define two additional sublevel sets to match the intermediate steps from the extensive-form problem to the Benders master problem made in Section 3.

$$\mathrm{S}_{EV}(\tau) = \mathrm{S}(\mathrm{G}_\theta, \mathrm{epi}(\mathrm{Q}), \tau) \tag{13}$$

$$\mathrm{S}_{PV}(\tau) = \mathrm{S}(\mathrm{g} + \mathrm{Q}, X, \tau) \tag{14}$$

The following lemma shows that the relationship between the epigraphical variant problem and Benders master problem sublevel sets is one of containment.

Lemma A.1. $S_{EV}(\tau) \subseteq S_{BM(\hat{V})}(\tau), \forall \hat{V} \subseteq V(\Pi), \forall \tau \in \mathbb{R}$

Proof. Proof This holds trivially if $\mathrm{S}_{EV}(\tau) = \varnothing$. Let $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{S}_{EV}(\tau)$ for a given $\tau \in \mathbb{R}$. To establish $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{S}_{BM(\hat{V})}(\tau)$, $\forall \hat{V} \subseteq V(\Pi)$ we need to show $\mathrm{g}(\bar{\boldsymbol{x}}) + \bar{\theta} \leq \tau$ and $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q}_{\hat{V}})$, $\forall \hat{V} \subseteq V(\Pi)$. We know that $\mathrm{g}(\bar{\boldsymbol{x}}) + \bar{\theta} \leq \tau$ holds by definition of $\mathrm{S}_{EV}(\tau)$. Since $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q})$ by definition of $\mathrm{S}_{EV}(\tau)$, $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q}_{\hat{V}})$, $\forall \hat{V} \subseteq V(\Pi)$ holds by Proposition 4.2. So $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{S}_{BM(\hat{V})}(\tau)$, $\forall \hat{V} \subseteq V(\Pi)$ and completes the containment proof. □

The relationship between the projected variable problem and the epigraphical variant problem sublevel sets is equivalence when projected into the shared first-stage variables. The intuition here is that the projected variable problem and the epigraphical variant problem are effectively the same problem with θ serving as the helper variable.

Lemma A.2. $S_{PV}(\tau) = proj_{\mathbb{R}^{n_1}}(S_{EV}(\tau)), \forall \tau \in \mathbb{R}$

Proof. Proof We prove this by double containment for fixed $\tau \in \mathbb{R}$. First, we show $\mathrm{S}_{PV}(\tau) \subseteq \mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EV}(\tau))$. We take $\bar{\boldsymbol{x}} \in \mathrm{S}_{PV}(\tau)$ and let $\bar{\theta} = \mathrm{Q}(\bar{\boldsymbol{x}})$. We need to show that $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q})$ and $\mathrm{g}(\bar{\boldsymbol{x}}) + \bar{\theta} \leq \tau$ to satisfy the definition of $\mathrm{S}_{EV}(\tau)$. Since $\bar{\boldsymbol{x}} \in X$ by definition of $\mathrm{S}_{PV}(\tau)$ and $\bar{\theta} = \mathrm{Q}(\bar{\boldsymbol{x}}) \in \mathbb{R}$ by the relatively complete recourse and dual non-emptiness assumptions on Q, this gives $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{epi}(\mathrm{Q})$. We know $\mathrm{g}(\bar{\boldsymbol{x}}) + \mathrm{Q}(\bar{\boldsymbol{x}}) \leq \tau$ by definition of $\mathrm{S}_{PV}(\tau)$, giving $\mathrm{g}(\bar{\boldsymbol{x}}) + \bar{\theta} \leq \tau$. Thus $\bar{\boldsymbol{x}} \in \mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EV}(\tau))$. Second, we show that $\mathrm{proj}_{\mathbb{R}^{n_1}}(\mathrm{S}_{EV}(\tau)) \subseteq \mathrm{S}_{PV}(\tau)$. We take $(\bar{\boldsymbol{x}}, \bar{\theta}) \in \mathrm{S}_{EV}(\tau)$. We need to show that $\bar{\boldsymbol{x}} \in X$ and $\mathrm{g}(\bar{\boldsymbol{x}}) + \mathrm{Q}(\bar{\boldsymbol{x}}) \leq \tau$ to establish $\bar{\boldsymbol{x}} \in \mathrm{S}_{PV}(\tau)$. We know $\bar{\boldsymbol{x}} \in X$ by definition of $\mathrm{S}_{EV}(\tau)$ as $\mathrm{proj}_{\mathbb{R}^{n_1}}(Q) \subseteq X$. As $\mathrm{g}(\bar{\boldsymbol{x}}) + \bar{\theta} \leq \tau$ and $\bar{\theta} \geq \mathrm{Q}(\bar{\boldsymbol{x}})$ by definition of $\mathrm{S}_{EV}(\tau)$, then $\mathrm{g}(\bar{\boldsymbol{x}}) + \mathrm{Q}(\bar{\boldsymbol{x}}) \leq \tau$ follows immediately. Thus $\bar{\boldsymbol{x}} \in \mathrm{S}_{PV}(\tau)$. □

The combination of Lemma A.1 and Lemma A.2 also gives the following corollary relating the projected variable problem and the Benders master problem sublevel sets:

Lemma A.3. $S_{PV}(\tau) \subseteq proj_{\mathbb{R}^{n_1}}(S_{BM(\hat{V})}(\tau)), \forall\ \hat{V} \subseteq V(\Pi), \forall \tau \in \mathbb{R}$

Proof. Proof This follows directly by the definition of the projection operator of (1) on Lemma A.1 and Lemma A.2. □

The relationship between the extensive-form problem and the projected variable problem sublevel sets is equivalence once projected to the shared first-stage variables. The intuition here is that the extensive-form problem and the projected variable problem are solving the same problem with the value function wrapping the second-stage elements of the extensive-form problem.

Lemma A.4. $proj_{\mathbb{R}^{n_1}}(S_{EF}(\tau)) = S_{PV}(\tau), \forall \tau \in \mathbb{R}$

Proof. Proof We prove this by double containment for a given $\tau \in \mathbb{R}$. First, we show that $\text{proj}_{\mathbb{R}^{n_1}}(\text{S}_{EF}(\tau)) \subseteq \text{S}_{PV}(\tau)$. We take $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \in \text{S}_{EF}(\tau)$. We need to show that $\bar{\boldsymbol{x}} \in X$ and $\text{g}(\bar{\boldsymbol{x}})+\text{Q}(\bar{\boldsymbol{x}}) \leq \tau$ to establish $\bar{\boldsymbol{x}} \in \text{S}_{PV}(\tau)$. We know that $\bar{\boldsymbol{x}} \in X$ by definition of $\text{S}_{EF}(\tau)$ as $\text{proj}_{\mathbb{R}^{n_1}}(\mathit{\Gamma}) \subseteq X$. We know that $\boldsymbol{q}^T\bar{\boldsymbol{y}} \geq \text{Q}(\bar{\boldsymbol{x}})$ for $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \in \mathit{\Gamma}$ by definition of Q and $\text{g}(\bar{\boldsymbol{x}}) + \boldsymbol{q}^T\bar{\boldsymbol{y}} \leq \tau$ by definition of $\text{S}_{EF}(\tau)$ giving $\text{g}(\bar{\boldsymbol{x}}) + \text{Q}(\bar{\boldsymbol{x}}) \leq \tau$. Thus $\bar{\boldsymbol{x}} \in \text{S}_{PV}(\tau)$. Second, we show that $\text{S}_{PV}(\tau) \subseteq \text{proj}_{\mathbb{R}^{n_1}}(\text{S}_{EF}(\tau))$. We take $\bar{\boldsymbol{x}} \in \text{S}_{PV}(\tau)$ and choose $\bar{\boldsymbol{y}} \in \text{argmin}_{\boldsymbol{y} \in \mathbb{R}^{n_2}_+} \boldsymbol{q}^T\boldsymbol{y}$ $s.t. \mathcal{W}\boldsymbol{y} + \mathcal{T}\boldsymbol{x} = \boldsymbol{h}$ which is guaranteed to exist by the relatively complete recourse assumption. We need $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \in \mathit{\Gamma}$ and $\text{g}(\bar{\boldsymbol{x}}) + \boldsymbol{q}^T\bar{\boldsymbol{y}} \leq \tau$ to show $\bar{\boldsymbol{x}} \in \text{proj}_{\mathbb{R}^{n_1}}(\text{S}_{EF}(\tau))$. We know $\bar{\boldsymbol{x}} \in X$ by definition of $\text{S}_{PV}(\tau)$ and $\bar{\boldsymbol{y}}$ was chosen to guarantee $\mathcal{W}\bar{\boldsymbol{y}} + \mathcal{T}\bar{\boldsymbol{x}} = \boldsymbol{h}$ hence $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \in \mathit{\Gamma}$. We know that $\boldsymbol{q}^T\bar{\boldsymbol{y}} = \text{Q}(\bar{\boldsymbol{x}})$ by choice of $\bar{\boldsymbol{y}}$ and $\text{g}(\bar{\boldsymbol{x}}) + \text{Q}(\bar{\boldsymbol{x}}) \leq \tau$ by definition of $\text{S}_{PV}(\tau)$ giving $\text{g}(\bar{\boldsymbol{x}}) + \boldsymbol{q}^T\bar{\boldsymbol{y}} \leq \tau$. Thus $\bar{\boldsymbol{x}} \in \text{proj}_{\mathbb{R}^{n_1}}(\text{S}_{EF}(\tau))$. □

B Non-Equivalence of the Extensive Form and Benders Sublevel Sets

In the previous section, we showed that the Benders master problem sublevel set, for fixed τ, contains the extensive-form sublevel set when both are projected to the shared first-stage variables. Ideally the two sublevel sets would be equivalent, which would enable the two sets to be used interchangeably. This is not the case in general as we show by providing a simple counterexample:

$$\begin{aligned} \text{Q}(x) = \min_{\boldsymbol{y}} & [1, \quad 1]\, y \\ & [1, \quad -1]\, \boldsymbol{y} = \boldsymbol{x} \\ & \boldsymbol{y} \in \mathbb{R}^2_+ \end{aligned}$$

This corresponds to $\mathcal{W} = [1, \quad -1]$, $q^T = [1, \quad 1]$, $T = -1$ and $\boldsymbol{h} = 0$. This results in $\mathit{\Pi} = [-1, 1]$ and $V(\mathit{\Pi}) = \{-1, 1\}$. The possible non-empty combination of vertices are as $V_1 = \{1\}$, $V_2 = \{-1\}$, and $V_3 = \{-1, 1\}$.

As approximations of Q, we see $\text{Q}_{\hat{V}_1}(x) = x$ and $\text{Q}_{\hat{V}_2}(x) = -x$. We recover $\text{Q}(x) = |x|$ with $\text{Q}_{\hat{V}_3}$ using both dual vertices, which was shown visually in Figure 1. We start with the relationship between approximate and exact epigraphs of the value function:

Remark B.1. *$epi(Q_{\hat{V}}) \subseteq epi(Q)$ does not hold in general.*

Proof. Proof Proof by construction. Let $\mathcal{W} = [1, \quad -1]$, $\boldsymbol{q}^T = [1, \quad 1]$, $T = -1$ and $h = 0$. Then $\text{Q}(x) = |x|$. Choose $\hat{V} = \{1\}$, so $\text{Q}_{\hat{V}}(x) = x$. $(-1, 0) \in \text{epi}(\text{Q}_{\hat{V}})$ but $(-1, 0) \notin \text{epi}(\text{Q})$. □

If the approximation of the value function is missing the supporting hyperplanes then there can be points admitted to the resulting approximate epigraph that the exact epigraph does not contain. Since the sublevel sets for the Benders master problem rely on the approximate epigraph, $\text{epi}(\text{Q}_{\hat{V}})$, it makes sense that the same example serves as a counterexample for the corresponding sublevel set relationships:

Remark B.2. *The converses of Lemma A.1, Lemma A.3, and Theorem 4.1 do not hold in general*

Proof. Proof Proof by construction. Let $\mathcal{W} = [1, \quad -1]$, $\boldsymbol{q}^T = [1, \quad 1]$, $T = -1$, $h = 0$ and and $\text{g}(x) = 0$. Then $\text{Q}(x) = |x|$ and $\text{g}(x) + \text{Q}(x) = |x|$. So $\text{S}_{EV}(-1) = \varnothing$, $\text{S}_{PV}(-1) = \varnothing$, and $\text{S}_{EF}(-1) = \varnothing$. Let $\hat{V} = \{1\}$ then $\text{Q}_{\hat{V}}(x) = x$ and $\text{g}(x) + \text{Q}_{\hat{V}}(x) = x$. Then $(-1, 0) \in \text{S}_{BM(\hat{V})}(-1)$, but $(-1, 0) \notin \text{S}_{EV}(-1)$, $\text{proj}_{\mathbb{R}^{n_1}}((-1, 0)) = -1 \notin \text{S}_{PV}(-1)$, and $\text{proj}_{\mathbb{R}^{n_1}}((-1, 0)) = -1$, $-1 \notin \text{proj}_{\mathbb{R}^{n_1}}(\text{S}_{EF}(-1))$. □

References

Ahanor I, Medal H, Trapp AC (2024) DiversiTree: A new method to efficiently compute diverse sets of near-optimal solutions to mixed-integer optimization problems. *INFORMS Journal on Computing* 36(1):61–77.

AIMMS (2024) AIMMS Documentation. URL `https://documentation.aimms.com/index.htmlx`.

Balas E, Jeroslow R (1972) Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics* 23(1):61–69.

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(1):238–252.

Bertsekas DP, Nedić A, Ozdaglar AE (2013) *Convex analysis and optimization* (Athena Scientific).

Birge JR, Louveaux F (2011) *Introduction to Stochastic Programming* (Springer Science & Business Media).

Brill E, Flach J, Hopkins L, Ranjithan S (1990) Mga: a decision support system for complex, incompletely defined problems. *IEEE Transactions on Systems, Man, and Cybernetics* 20(4):745–757, ISSN 2168-2909.

Brill ED (1979) The use of optimization models in public-sector planning. *Management Science* 25(5):413–422.

Brill ED, Chang SY, Hopkins LD (1982) Modeling to generate alternatives: The hsj approach and an illustration using a problem in land use planning. *Management Science* 28(3):221–235.

Brown G, Carlyle M, Salmerón J, Wood K (2006) Defending critical infrastructure. *Interfaces* 36(6):530–544.

Byeon G, Van Hentenryck P (2022) Benders subproblem decomposition for bilevel problems with convex follower. *INFORMS Journal on Computing* 34(3):1749–1767.

Byers TH, Waterman MS (1984) Technical note—determining all optimal and near-optimal solutions when solving shortest path problems by dynamic programming. *Operations Research* 32(6):1381–1384.

Bynum ML, Hackebeil GA, Hart WE, Laird CD, Nicholson BL, Siirola JD, Watson JP, Woodruff DL (2021) *Pyomo–optimization modeling in python*, volume 67 (Springer Science & Business Media), third edition.

Conforti M, Cornuéjols G, Zambelli G (2014) *Integer Programming*. Graduate Texts in Mathematics (Springer International Publishing), ISBN 9783319110080.

Danna E, Fenelon M, Gu Z, Wunderling R (2007) Generating multiple solutions for mixed integer programming problems. Fischetti M, Williamson DP, eds., *Integer Programming and Combinatorial Optimization*, 280–294 (Berlin, Heidelberg: Springer Berlin Heidelberg), ISBN 978-3-540-72792-7.

Eckstein J, Hart WE, Phillips CA (2015) Pebbl: an object-oriented framework for scalable parallel branch and bound. *Mathematical Programming Computation* 7:429–469.

Elçi O, Hooker J (2022) Stochastic planning and scheduling with logic-based benders decomposition. *INFORMS Journal on Computing* 34(5):2428–2442.

Gurobi Optimization, LLC (2024) Gurobi Optimizer Reference Manual.

Hart WE, Johnson ES, Phillips CA, Aquino A, Ammari B, Arguello B, Davis SA, Gearhart JL, Laird CD, Mattes CL, et al. (2024) CI-MOR Final Report: Analysis and Validation of Critical Infrastructure Models using Model Order Reduction. Technical report, Sandia National Lab. (SNL-NM), Albuquerque, NM (United States).

IBM ILOG CPLEX (2024) User's Manual for CPLEX.

Israeli E, Wood RK (2002) Shortest-path network interdiction. *Networks* 40(2):97–111.

Larsen E, Lachapelle S, Bengio Y, Frejinger E, Lacoste-Julien S, Lodi A (2022) Predicting tactical solutions to operational planning problems under imperfect information. *INFORMS Journal on Computing* 34(1):227–242.

Lau M, Patankar N, Jenkins JD (2024) Measuring exploration: evaluation of modelling to generate alternatives methods in capacity expansion models. *Environmental Research: Energy* 1(4):045004.

Lee S, Phalakornkule C, Domach MM, Grossmann IE (2000) Recursive milp model for finding all the alternate optima in lp models for metabolic networks. *Computers & Chemical Engineering* 24(2-7):711–716.

Paris Q (1981) Multiple optimal solutions in linear programming models. *American Journal of Agricultural Economics* 63(4):724–727.

Paris Q (1983a) Multiple optimal solutions in linear programming models: Reply. *American Journal of Agricultural Economics* 65(1):184–186.

Paris Q (1983b) Multiple optimal solutions in quadratic programming models. *Western Journal of Agricultural Economics* 8(2):141–154.

Paris Q (2010) *Economic foundations of symmetric programming* (Cambridge University Press).

Pedersen TT, Victoria M, Rasmussen MG, Andresen GB (2021) Modeling all alternative solutions for highly renewable energy systems. *Energy* 234:121294.

Petit T, Trapp AC (2019) Enriching solutions to combinatorial problems via solution engineering. *INFORMS Journal on Computing* 31(3):429–444.

Robinson SM (1996) Analysis of sample-path optimization. *Mathematics of Operations Research* 21(3):513–528.

Rockafellar RT (1996) *Convex analysis (Princeton paperbacks)* (Princeton University Press).

Van Slyke RM, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* 17(4):638–663.

Williams HP (2013) *Model building in mathematical programming* (Wiley).