A Method of Constructing the Efficient Frontier by Exploiting the Value Function (VF)

Samira Fallah
saf418@lehigh.edu

Ted K. Ralphs
ted@lehigh.edu

Lawrence V. Snyder
lvs2@lehigh.edu

INDUSTRIAL AND SYSTEMS ENGINEERING

LEHIGH UNIVERSITY

October 26, 2021

Motivation

- We are interested in the relationship between the VF and the multi-obj efficient frontier. The value function and the set of non-dominated points by efficient frontier have the same information but are presented in two different ways.
- Considered an approach for constructing the efficient frontier for a general multi-objective Mixed Integer Linear Programming (MILP) problem by exploiting its relationship to the VF.
- There is an existing algorithm[1] for constructing the full VF of a general MILP with all varying RHS, but we are interested in the case in which we have some fixed RHS.
- We are targeting the applications in which the frontier is a constraint (such as bilevel optimization).

[1] A. Hassanzadeh and T.K. Ralphs. *On the Value Function of a Mixed Integer Linear Optimization Problem and an Algorithm for Its Construction*. Tech. rep. COR@L Laboratory Report 14T-004, Lehigh University, 2014. URL: http://coral.ie.lehigh.edu/~ted/files/papers/MILPValueFunction14.pdf.

The relationship between the VF and the efficient frontier

Figure: The Value Function

$$
\begin{aligned}
z(b) = \inf \quad & c_I^T x_I + c_C^T x_C \\
& d_I^T x_I + d_C^T x_C = b \\
& {d'}_I^T x_I + {d'}_C^T x_C = b' \\
& x_I \in \mathbb{Z}_+^r, \;\; x_C \in \mathbb{R}_+^{n-r}.
\end{aligned}
$$

Figure: The bi-objective problem

$$
\begin{aligned}
\inf \quad & (c_I^T x_I + c_C^T x_C, d_I^T x_I + d_C^T x_C) \\
& {d'}_I^T x_I + {d'}_C^T x_C = b' \\
& x_I \in \mathbb{Z}_+^r, \;\; x_C \in \mathbb{R}_+^{n-r}.
\end{aligned}
$$

- The efficient frontier is given by points of the form

$$(c_I^T x_I + c_C^T x_C, d_I^T x_I + d_C^T x_C) \tag{1}$$

where $(x_I, x_C) \in (\mathbb{Z}_+^r \times \mathbb{R}_+^{n-r})$ is a non-dominated point (NDP).

- The graph of the value function is also given by the same pair (1). Every NDP from the efficient frontier set gives a point on the value function graph. However, not all points on the value function graph are NDP.

Numerical example to show the relationship

$$
\begin{aligned}
z(b) = \inf \quad & 2x_1 + 2x_2 + 5x_3 + 6x_5 + 3x_6 + 6x_7 + 7y_2 + 10y_3 + 2y_4 + 10y_5 \\
& -x_1 + 3x_2 - 9x_3 - 3x_5 + 9x_6 + 2x_7 + 10y_1 + 8y_2 + y_3 - 7y_4 + 6y_5 = b \\
& -x_1 + 10x_3 + 5x_4 + x_5 + 4x_6 - 3x_7 + 9y_1 + 3y_2 + 2y_3 + 6y_4 - 10y_5 = 4 \\
& x_i \in \mathbb{Z}_+ \quad \forall i \in \{1, 2, \dots, 7\} \\
& y_i \in \mathbb{R}_+ \quad \forall i \in \{1, 2, \dots, 5\}
\end{aligned}
$$

Figure: The Value Function

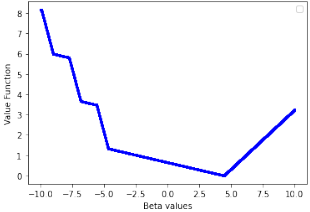
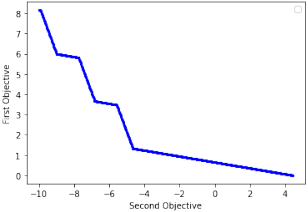


Figure: The Efficient Frontier



- So when we have two minimization objectives, the efficient frontier is exactly the non-increasing part of the value function.

Problem Definition

Let the MILP problem as:

$$z_{\text{MILP}} = \inf_{(x_I, x_C) \in X} \quad c_I^T x_I + c_C^T x_C,$$

where

$$X = \{(x_I, x_C) \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} : A_I x_I + A_C x_C = d, A_I' x_I + A_C' x_C = d'\}$$

The VF of the MILP above:

$$z(b) = \inf_{(x_I, x_C) \in S(b)} \quad c_I^T x_I + c_c^T x_c,$$

where

$$S(b) = \{(x_I, x_C) \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} : A_I x_I + A_C x_C = b, A_I' x_I + A_C' x_C = b'\}$$

$$S_I(b) = \{x_I \in \mathbb{Z}_+^r : A_I x_I + A_C x_C = b, A_I' x_I + A_C' x_C = b'\}$$

$$S_I = \cup_{b \in B} S_I(b)$$

Note that b' is fixed with the associated dimension.

Structure of the MILP VF

Let's consider Continuous Restriction (CR) w.r.t a given $\hat{x}_I$

$$\begin{aligned} \bar{z}(b;\hat{x}_I) = \quad & c_I^T\hat{x}_I + \text{inf } c_C^T x_C \\ & A_C x_C = b - A_I\hat{x}_I \\ & A'_C x_C = b' - A'_I\hat{x}_I \\ & x_C \in \mathbb{R}_+^{n-r} \end{aligned}$$

Let $S(b,\hat{x}_I) = \{x_C \in \mathbb{R}_+^{n-r} : A_C x_C = b - A_I\hat{x}_I,\ A'_C x_C = b' - A'_I\hat{x}_I\}$.
When $\hat{x}_I = 0$, the VF has the same property as the VF of a general LP, since we have:

$$\begin{aligned} z_C(b) = \text{inf} \quad & c_C^T x_C \\ & A_C x_C = b \\ & A'_C x_C = b' \\ & x_C \in \mathbb{R}_+^{n-r} \end{aligned}$$

Main property of the MILP value function

Proposition

For any $\hat{x} \in S_I$, $\bar{z}(.;\hat{x})$ bounds z from above:

$$\bar{z}(b;\hat{x}) = c_I^T\hat{x} + z_C(b - A_I\hat{x}) \geq \textit{inf}_{x \in S_I}\ c_I^T x + z_C(b - A_I x) = z(b)$$

Note that the set of RHS, $b \in B$, is bounded.

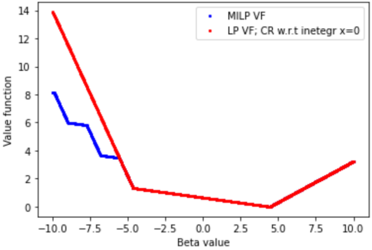
Based on the proposition:

$$z(b) = \inf_{x_I \in S_I} \bar{z}(b; x_I),$$

So the MILP VF is the minimum of translated slices of the full LP VF.

The MILP VF vs restricted LP VF

As we can see the MILP VF is a number of translations of the restricted LP.



Proposed Algorithm

Algorithm 1: Value Function Algorithm for a general MILP

Input: $\bar{z}(b) = \infty$ for all $b \in B$, $\Gamma^0 = \infty$, $k = 0$,
set x_I^0 as the optimal solution of $\min c_I^T x_I + c_C^T x_C$ where $A'_I x_I + A'_C x_C = b'$,
$x_I \in \mathbb{Z}_+^r, x_C \in \mathbb{R}_+^{n-r}$ and set $S^0 = \{x_I^0\}$.
Output: $z(b) = \bar{z}(b) \quad \forall b \in B$.

```
while Γ^k > 0 do
    Let z̄(b) = min{z̄(b), z̄(b; x_I^k)} for all b ∈ B.
    k ← k + 1.
    Solve  Γ^k = max{z̄(b) − c_I^T x_I − c_C^T x_C} s.t. A_I x_I + A_C x_C = b,
        A'_I x_I + A'_C x_C = b', x_I ∈ Z_+^r, x_C ∈ R_+^{n−r}, to obtain x_I^k.
    Set S^k ← S^{k−1} ∪ {x_I^k}.
end
```

So the algorithm only collects the integer parts for constructing the VF and this is all that is needed for many applications.

Solving the subproblem (SP)

The subproblem arises in the algorithm can be formulated as a Mixed Integer Nonlinear Programming (MINLP) as follows:

$$\begin{array}{ll} \Gamma^k = \max & \bar{z}(b) - c_I^T x_I - c_C^T x_C \\ \text{subject to} & A_I x_I + A_C x_C = b \\ & A'_I x_I + A'_C x_C = b' \\ & x_I \in \mathbb{Z}_+^r, x_C \in \mathbb{R}_+^{n-r} \end{array}$$

For practical purpose the subproblem can be written as:

$$\begin{array}{ll} \Gamma^k = \max & \theta \\ \text{subject to} & \theta \leq \bar{z}(b) - c_I^T x_I - c_C^T x_C \\ & A_I x_I + A_C x_C = b \\ & A'_I x_I + A'_C x_C = b' \\ & x_I \in \mathbb{Z}_+^r, x_C \in \mathbb{R}_+^{n-r} \end{array}$$

We know that in iteration $k \geq 1$ of the algorithm we have

$$\bar{z}(b) = \min_{i=0,\ldots,k-1} \bar{z}(b; x_I^i),$$

Solving the SP - Cont.

therefore we have:

$$\theta + c_I^T x_I + c_C^T x_C \leq c_I^T x_I^i + z_C(b - A_I x_I^i, b' - A'_I x_I^i) \quad \forall i \in \{0, \ldots, k-1\}$$

Next, we can write z_C as:

$$z_C(b - A_I x_I^i, b' - A'_I x_I^i) = \sup\{(b - A_I x_I^i)^T v^i + (b' - A'_I x_I^i)^T v'^i : \\ A_C^T v^i + A'^T_C v'^i \leq c_C, (v^i, v'^i) \in (\mathbb{R}^m \times \mathbb{R}^{m'})\}$$

Then reformulate each of k constraints as

$$\begin{aligned} &\theta + c_I^T x_I + c_C^T x_C \leq c_I^T x_I^i + (b - A_I x_I^i)^T v^i + (b' - A'_I x_I^i)^T v'^i \\ &A_C^T v^i + A'^T_C v'^i \leq c_C \\ &(v^i, v'^i) \in \mathbb{R}^m \times \mathbb{R}^{m'} \end{aligned}$$

Together, in each iteration we solve

$$\begin{array}{ll} \Gamma^k = \max & \theta \\ \text{subject to} & \theta + c_I^T x_I + c_C^T x_C \leq c_I^T x_I^i + (A_I x_I + A_C x_C - A_I x_I^i)^T v^i + \\ & \qquad +(b' - A'_I x_I^i)^T v'^i \quad \forall i \in \{0, \ldots, k-1\} \\ & A_C^T v^i + A'^T_C v'^i \leq c_C \quad \forall i \in \{0, \ldots, k-1\} \\ & A'_I x_I + A'_C x_C = b' \\ & (v^i, v'^i) \in \mathbb{R}^m \times \mathbb{R}^{m'} \quad \forall i \in \{0, \ldots, k-1\} \\ & x_I \in \mathbb{Z}^r_+, \; x_C \in \mathbb{R}^{n-r}_+ \\ & \theta \in \mathbb{R} \end{array}$$

Note that we can add some slack variables to our reformulation to get the efficient frontier part of the VF, but it is optional.

Conclusions

- We presented an algorithm for constructing the efficient frontier for a general MILP with some fixed RHSs.
- To the best of our knowledge, this algorithm is being presented for the first time in literature.
- One potential advantage is the algorithm gives a performance guarantee if we stop early.
- We are implementing the MINLP algorithm in Python using the nonlinear solver *Couenne*.
- We will perform an extensive computational experiment later to evaluate the effectiveness of the proposed algorithm.

Future Work

- Since the problem is nonlinear, we are targeting a customized algorithm.
- We can develop a warm start heuristic algorithm for solving the problem at each iteration.
- The bilinear terms can be linearized with McCormick inequalities when all integer variables are binary.

Questions?

Thank you!