Approximating the Stability Region of Binary Mixed-Integer Programs

Fatma Kılınç Karzan, Alejandro Toriello, Shabbir Ahmed, George Nemhauser, Martin Savelsbergh
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology

Summary

We consider optimization problems with some binary variables, where the objective function is linear in these variables. The *stability region* of a given solution of such a problem is the polyhedral set of objective coefficients for which the solution is optimal. A priori knowledge of this set provides valuable information for sensitivity analysis and re-optimization when there is objective coefficient uncertainty. An exact description of the stability region typically requires an exponential number of inequalities. We develop useful polyhedral inner and outer approximations of the stability region using only a linear number of inequalities. Furthermore, when a new objective function is not in the stability region, we produce a list of good solutions that can be used as a quick heuristic or as a warm start for future solves.

Stability Region

Consider the optimization problem

$$\max \ \{c^*x + h(y) : (x, y) \in X, \ x \in \{0,1\}^n\}, \qquad (P(c^*))$$

where $c^*x = \sum_{i \in N} c_i^* x_i$ and $N = \{1, \ldots, n\}$. By possibly complementing the x variables, we assume without loss of generality that (x^*, y^*) is an optimal solution with $x^* = 0$ and objective value $z^* = c^*x^* + h(y^*) = h(y^*)$. We are interested in the stability of (x^*, y^*) with respect to perturbations of c^*.

Stability Region of (x^*, y^*): Region $C \subseteq \mathbb{R}^n$ s.t. $c \in C$ iff (x^*, y^*) is optimal for $P(c)$, i.e.,

$$C = \{c \in \mathbb{R}^n : cx \leq h(y^*) - h(y), \forall (x, y) \in X\}.$$

Remark 1 *Let $\hat{x} \in \{0,1\}^n$, and define the optimization problem*

$$v(\hat{x}) = \max \{h(y) : (\hat{x}, y) \in X\}.$$

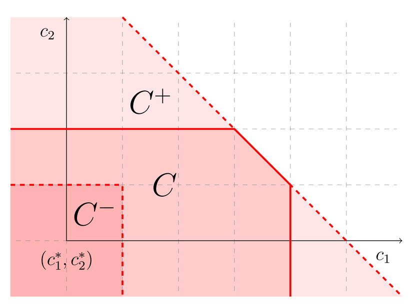
If $\hat{x} \notin \mathrm{proj}_x(X) = \{x \in \{0,1\}^n : \exists\, y \text{ s.t. } (x, y) \in X\}$, define $v(\hat{x}) = -\infty$. Then

$$C = \{c \in \mathbb{R}^n : cx \leq h(y^*) - v(x), \forall x \in \{0,1\}^n\}.$$

Approximating C

Inner Approximation Neighborhood, C^-: Region $C^- \subseteq \mathbb{R}^n$ s.t. (x^*, y^*) is optimal for $P(c), \forall c \in C^-$.

Outer Approximation Neighborhood, C^+: Region $C^+ \subseteq \mathbb{R}^n$ s.t. (x^*, y^*) is not optimal for $P(c), \forall c \notin C^+$.



Proposition 1

- $C^- \subseteq C \subseteq C^+$.
- *$\{c \in \mathbb{R}^n : c \leq c^*\}$ is an inner neighborhood of (x^*, y^*).*
- *$\mathrm{conv}(C_1^- \cup C_2^-)$ is an inner neighborhood.*
- *$C_1^+ \cap C_2^+$ is an outer neighborhood.*

Lemma 1 *Let $(\hat{x}, \hat{y})$ be feasible for $P(c^*)$, with objective value $\hat{z} = c^*\hat{x} + h(\hat{y}) \leq z^*$. Suppose $\hat{c} \in \mathbb{R}^n$ satisfies*

$$\hat{c}\hat{x} > z^* - \hat{z} + c^*\hat{x}.$$

Then (x^*, y^*) is not optimal for $P(\hat{c})$.

Algorithm

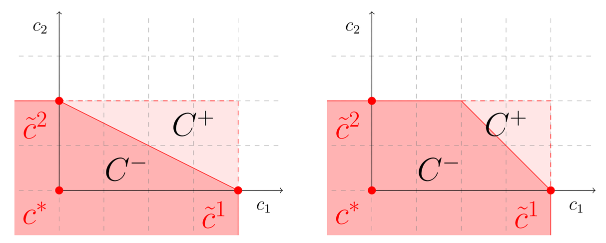
We now consider the restricted problem:

$$z_j = \max \ \{c^*x + h(y) : (x, y) \in X, \ x \in \{0,1\}^n, \ x_j = 1\} \qquad (P_j)$$

W.l.o.g. we assume that P_j is feasible. Let (x^j, y^j) be an optimal solution to P_j and define $\gamma_j \equiv z^* - z_j + c_j^*$.

Lemma 2 *Let $\tilde{c}^j = (c_1^*, \ldots, c_{j-1}^*, \gamma_j, c_{j+1}^*, \ldots, c_n^*)$. The solution (x^*, y^*) is optimal for $P(\tilde{c}^j)$, $\forall j \in N$.*

By solving each of the problems P_j in turn, we have a list of cost vectors $\{\tilde{c}^j\}_{j \in N}$ for which we have proved the optimality of (x^*, y^*). These points together with c^* form the vertices of a simplex, which by Proposition 1 is an inner neighborhood of (x^*, y^*) (see Figure a.) However, from Remark 1, we also know that every inequality in the description of the stability region is of the form $cx \leq h(y^*) - v(x)$, for some $x \in \mathrm{proj}_x(X)$. We can therefore exploit the known structure of these inequalities to increase the inner neighborhood, as in Figure b. Both regions can be expanded by adding the negative orthant to every point, as explained in Corollary 1.



(a) (b)

The following theorem formalizes this notion.

Theorem 1 *Suppose we order the x variables so that $z^* \geq z_1 \geq z_2 \geq \cdots \geq z_n$ holds. Then*

$$C^- = \left\{c \geq c^* : \sum_{i=1}^{j} c_i \leq \gamma_j + \sum_{i=1}^{j-1} c_i^*, \forall j \in N\right\}$$

is an inner neighborhood of (x^*, y^*).

Corollary 1 *The set $\{c + d : c \in C^-, d \leq 0\}$ is an inner neighborhood of (x^*, y^*).*

These last two results motivate a natural algorithm for determining an inner neighborhood. Solve each of the problems P_j in turn, sort them by objective value, and compute the inner neighborhood as indicated in Theorem 1. As we shall see next, this procedure can be modified slightly to potentially reduce the number of solves while still yielding the same region.

Outline of the Algorithm

Require: Problem $P(c^*)$ with optimal solution (x^*, y^*) satisfying $x^* = 0$.

Set $(x^0, y^0) \leftarrow (x^*, y^*)$, $z_0 \leftarrow z^*$, $I \leftarrow N$.
Add cut $D \equiv (\sum_{i \in I} x_i \geq 1)$ to $P(c^*)$.
for $k = 1, \ldots, n$ **do**
  Resolve modified $P(c^*)$; get new optimal solution (x^k, y^k) and objective value $c^*x^k + h(y^k) = z_k$.
  if $P(c^*)$ is infeasible **then**
    Set $I_\infty \leftarrow I$.
    Return k and exit.
  end if
  Set $I_k^+ \leftarrow \{i \in N : x_i^k = 1\}$, $I_k^- \leftarrow I \cap I_k^+$.
  Set $I \leftarrow I \setminus I_k^-$; modify cut D accordingly.
end for

An outer neighborhood is easily obtained applying Lemma 1 to each solution (x^k, y^k).

Theorem 2 *The set $C^+ = \left\{c \in \mathbb{R}^n : cx^j \leq z^* - z_j + c^*x^j, \forall j \in N\right\}$ is an outer neighborhood of $P(c^*)$. The outer neighborhood C^+ satisfies*

$$\{c \in C^+ : c \geq c^*\} \subseteq \{c \in \mathbb{R}^n : c_j^* \leq c_j \leq \gamma_j, \forall j \in N\}.$$

To determine an inner neighborhood, we first establish a preliminary fact.

Lemma 3 *Let $j \in I_k^-$, for some k. Then (x^k, y^k) is optimal for P_j.*

Theorem 3 *Suppose Algorithm is run on problem $P(c^*)$, terminating after $\ell \leq n$ steps. Let $(x^k, y^k), z_k, I_k^+, I_k^-, \forall k = 1, \ldots, \ell$ and I_∞ be obtained from the algorithm. Then*

$$C^- = \left\{c \geq c^* : \sum_{i \in I_1^- \cup \cdots \cup I_k^-} c_i \leq z^* - z_k + \sum_{i \in I_1^- \cup \cdots \cup I_k^-} c_i^*, \forall k = 1, \ldots, \ell\right\}$$

is an inner neighborhood of $P(c^*)$.

Comments

The stability region C depends only on (x^*, y^*) and $h(y)$, not on c^*. However, the inner neighborhood C^- calculated using Algorithm does depend on both c^* and the list of solutions $\{(x^k, y^k)\}_{k=1}^{\ell}$ produced by Algorithm .

The list of solutions $\{(x^k, y^k)\}_{k=1}^{\ell}$ produced by Algorithm is also useful when (x^*, y^*) is *not* optimal for $P(c)$. Given a cost vector c, we can produce the best solution from the list as a quick heuristic or a warm start for future solves.

Computational Results

Our goal is to study the quality of the inner and outer approximations of the stability region, and to test the value of the solutions obtained during the execution of the algorithm when re-optimization is necessary. All computational experiments were carried out on a system with two 2.4 GHz Xeon processors and 2 GB RAM, and using CPLEX 9.0 as the optimization engine.

The set of instances contains pure and mixed-integer linear programs from MIPLIB 3.0. For each instance, we take all binaries as x variables, i.e., variables under scrutiny, and all others as y variables. The cost vector for x variables is randomly and uniformly perturbed within 1%, 2%, 5%, 10% and 20% of c_j^* for each j independently, *always in the most pertinent direction* $(c \geq c^*)$. 1000 random cost perturbations examined for each instance and percentage combination.

Tables 1 and 2 provide the results of our computational experiments for 5% perturbations.

	Region Counts			Times Best is Optimal		
Problem	C^-	$C^+ \setminus C^-$	$\mathbb{R}^n \setminus C^+$	$C^+ \setminus C^-$	$\mathbb{R}^n \setminus C^+$	Total
dcmulti	1000	0	0	-	-	1000
egout	783	217	0	217	-	1000
gen	0	0	1000	-	631	631
khb05250	360	640	0	627	-	987
lseu	0	0	1000	-	760	760
mod008	0	0	1000	-	80	80
modglob	0	1000	0	1000	-	1000
p0033	0	0	1000	-	8	8
p0201	0	0	1000	-	1000	1000
p0282	50	950	0	950	-	1000
pp08a	0	0	1000	-	393	393
qiu	0	0	1000	-	37	37
stein27	0	0	1000	-	0	0
vpm2	0	0	1000	-	315	315

Table 1: Region and optimality counts for 5% perturbations.

	$\frac{(\text{optimal}-\text{best})}{\lvert\text{optimal}\rvert}$		95% CI on Relative Diff.	
Problem	Average	St. Dev.	LB	UB
gen	1.06E-05	1.78E-05	1.05E-05	1.06E-05
khb05250	4.46E-06	3.81E-05	4.38E-06	4.53E-06
lseu	3.25E-04	8.06E-04	3.23E-04	3.26E-04
mod008	5.91E-03	3.89E-03	5.90E-03	5.91E-03
p0033	5.12E-03	1.96E-03	5.12E-03	5.12E-03
pp08a	3.03E-04	3.96E-04	3.02E-04	3.04E-04
qiu	3.49E-02	1.94E-02	3.49E-02	3.50E-02
stein27	6.68E-03	1.88E-03	6.67E-03	6.68E-03
vpm2	6.87E-04	8.07E-04	6.85E-04	6.89E-04

Table 2: Relative difference results for 5% perturbations, where relevant.

Comparing the Approximations to the Stability Region

Although we cannot efficiently compute the exact stability region of (x^*, y^*) when $P(c^*)$ is NP-hard, if the convex hull of the feasible region is given by polynomially many inequalities, as is the case when the constraint matrix is totally unimodular and the formulation has polynomial size, we can use well-known linear programming theory to generate the region. Specifically, suppose

$$x^* \in S = \{x \in \mathbb{R}^n : a^i x \leq b_i, \forall i = 1, \ldots, m\},$$

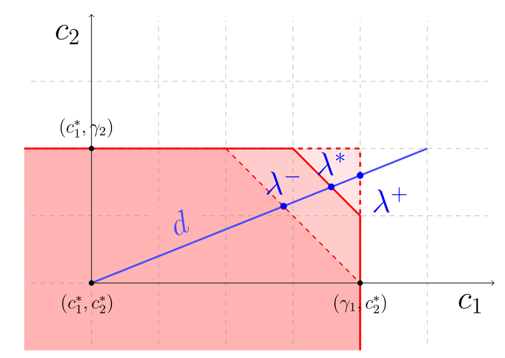
where $a^i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. Then by complementary slackness and LP duality, x^* is optimal for $\max\{cx : x \in S\}$ iff $c \in C = \mathrm{cone}\{a_i : i \in I^*\}$, where $I^* = \{i : a^i x^* = b_i\}$.

For our next set of experiments, we have chosen instances of the assignment problem. We generate four 20×20 specific problem instances using costs from problem `assign100`, originally from Beasley, 1990.

We generate a random direction vector d, where each component $d_i \backsim U(0,1)$ is an i.i.d. random variable. We calculate the value

$$\lambda^- = \max\{\lambda : c^* + \lambda d \in C^-\},$$

and similarly define and calculate λ^* and λ^+ for C and C^+, respectively. Note that we always have the relation $\lambda^- \leq \lambda^* \leq \lambda^+$, because $C^- \subseteq C \subseteq C^+$.



For each assignment instance, we examine 100,000 randomly generated d vectors. Our results are summarized in Table 3.

Problem	λ^- (avg.)	λ^* (avg.)	λ^+ (avg.)	$\frac{\lambda^*-\lambda^-}{\lambda^+-\lambda^-}$ (avg.)	$\lambda^* - \lambda^- > \lambda^+ - \lambda^*$ (ct.)	$\lambda^- = \lambda^+$ (ct.)
5×5						
assign20_1	6.0086	12.1997	12.2018	0.9999	62,835	37,164
assign20_2	4.2159	7.3348	7.3355	0.9999	98,325	1,674
assign20_3	5.2905	6.3897	6.4002	0.9975	23,673	76,326
assign20_4	3.8208	4.0776	4.078	0.9997	16,899	83,100
10×10						
assign20_1	1.2708	2.3431	2.3477	0.9982	99,995	4
assign20_2	1.5802	2.8811	2.8811	1	99,999	0
assign20_3	1.5977	1.8501	1.8501	1	45,613	54,386
assign20_4	1.4261	3.2234	3.2802	0.9789	99,996	2
20×20						
assign20_1	0.3669	0.9154	0.9246	0.9858	100,000	0
assign20_2	0.4758	0.7647	0.7647	1	100,000	0
assign20_3	0.2593	0.2613	0.2613	1	7,614	92,386
assign20_4	0.2397	0.3493	0.3494	0.9999	98,298	1,702

Table 3: Averages and counts for the assignment instances.

Conclusions

We have developed an algorithm to approximate stability region for binary MIPs which solves at most a *linear* number of closely related problems. Computational experiments demonstrate

- C^- and C^+ closely approximate the stability region.
- The list of solutions generated as a byproduct of our algorithm can be used effectively to produce high quality solutions for a true cost vector that is close to our estimate c^*.
- Results from the shooting experiments indicate that our inner approximation, C^-, can be too conservative, since it accounts for most of the uncertain region $C^+ \setminus C^-$.