**10-725/36-725: Convex Optimization** **Fall 2016**

# Lecture 25: Mixed Integer Programming (part 2)

*Lecturer: Javier Peña* *Scribes: Paul Liang, Xupeng Tong, Akash Bharadwaj*

This lecture's notes illustrate some uses of various LaTeX macros. Take a look at this and imitate.

## 25.1 Recap of Previous Lecture

Recall the integer programming problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & x \in C \\ & x_j \in \mathbb{Z}, j \in J \end{aligned}$$

where $f : \mathbb{R}^n \to R$, $C \subseteq \mathbb{R}^n$ are convex and $J \subseteq \{1, \cdots, n\}$. Basically a convex objective function with convex constraints, but some of the variables are integers. We ended with a general algorithm for solving Enumeration approach to solve the combinatorial aspect of the problem, but one which is not brute force. Rather a divide and conquer approach to increase efficiency.

### 25.1.1 Branch and Bound algorithm

Suppose the original program is called IP.

1. Solve the convex relaxation (CR) that ignores the integer constraint.

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & x \in C \end{aligned}$$

2. (CR) infeasible $\Rightarrow$ (IP) is infeasible. Stop
3. Solution $x^*$ to (CR) is (IP) feasible $\Rightarrow$ $x^*$ solves (IP). Stop
4. Solution $x^*$ to (CR) not (IP) feasible $\Rightarrow$. This happens when one of the components in $x^*$ is not integer when it is required to be integer. lower bound for (IP). Branch and recursively solve subproblems.

Given 2 convex relxations, the one with more constraints will result in a smaller feasible set. Which one should we prefer? The one with more constraints! With more constraints, we have a tigher convex relexation and the lower bound we get is also tighter. Bounding is critical for branch and bound, so generally the more constraints you can put in your model, the better.

## 25.2 Convexification

Consider the special case of an integer program with linear objective. We can always push the

$$\begin{aligned} \min_x \quad & c^T x \\ \text{subject to} \quad & x \in C \\ & x_j \in \mathbb{Z}, j \in J \end{aligned}$$

Now this program is equivalent to the following:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{subject to} \quad & x \in S \end{aligned}$$

where $S = \text{conv}\,\{x \in C : x_j \in \mathbb{Z}, j \in J\}$. And this becomes a convex optimization problem.

This equivalence is simply a consequence of linearity in the objective function.

Now we can apply this fact to our integer linear programs:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x_j \in \mathbb{Z}, j \in J \end{aligned}$$

**Theorem 25.1** *If $A, b$ are rational, then the set*

$$S = conv\,\{x : Ax \leq b, x_j \in \mathbb{Z}, j \in J\}$$

*is a polyhedron.*

Thus the above integer linear program is equivalent to a linear program. How hard can this be? Here's the catch: the polyhedron could have an exponential number of inequalities and be very complicated.

## 25.3 Cutting plane algorithm

We say that the inequality $\pi^T x \leq \pi_0$ is **valid** for a set $S$ if

$$\pi^T x \leq \pi_0 \text{ for all } x \in S.$$

Consider the problem (IP):

$$\begin{aligned} \min_x \quad & c^T x \\ \text{subject to} \quad & x \in C \\ & x_j \in \mathbb{Z}, j \in J \end{aligned}$$

In words, first solve the relaxation. If the relaxation didn't give a solution to the problem, that implies the solution found is non-integer, so we need to cut the solution set off with a valid inequality for your problem. Then resolve.

Cutting plane algorithm for problem (IP):

```
1. let $C_0 := C$ and compute $x^{(0)} := \text{argmin}_x\{c^Tx : x \in C_0\}$. If $x^{(0)}$ solves the problem then we are done.
2. for $k = 0, 1, \ldots$
       if $x^{(k)}$ is (IP) feasible then $x^{(k)}$ is an optimal solution. Stop.
       else
           find a valid inequality $(\pi, \pi_0)$ for $S$ that cuts off $x^{(k)}$.
           let $C_{k+1} := C_k\{x : \pi^T x \le \pi_0\}$
           compute $x^{(k+1)} := \text{argmin}_x\{c^Tx : x \in C_{k+1}\}$
       end if
   endfor
```

A valid inequality is also called a cutting plane or a cut.

## 25.4 Types of cuts

### 25.4.1 Gomory Cuts (1958)

This class of cuts is based on the following observation:

If $a \le b$ and a is an integer then $a \le \lfloor b \rfloor$.

Suppose

$$S \subseteq \left\{ x \in \mathbb{Z}^n_+ : \sum_{j=1}^n a_j x_j = a_0 \right\}$$

where $a_0 \notin \mathbb{Z}$. The Gomory fractional cut is:

$$\sum_{j=1}^n (a_j - \lfloor a_j \rfloor) x_j \ge a_0 - \lfloor a_0 \rfloor$$

### 25.4.2 Lift-and-project Cuts

**Theorem 25.2** *(Balas 1974) Assume $C = \{x : Ax \le b\} \subseteq \{x : 0 \le x_j \le 1\}$. Then $C_j := conv\{x \in C : x_j \in \{0, 1\}\}$ is the projection of the polyhedron $\mathcal{L}_j(C)$ defined by the following inequalities:*

$$\begin{array}{rcl} Ay & \le & \lambda b \\ Az & \le & (1-\lambda) b \\ y + z & = & x \\ \lambda & \le & 1 \\ \lambda & \ge & 0. \end{array}$$

Suppose $\tilde{x}$ C but $\tilde{x} \notin C_j$ . To find a cut separating $\tilde{x}$ from $C_j$, solve:

$$\begin{array}{ll} \min\limits_{x,y,z,\lambda} & \|x - \tilde{x}\| \\ \text{subject to} & (x, y, z, \lambda) \in \mathcal{L}_j(C) \end{array}$$

## 25.5 Branch and cut algorithm

Combine strengths of both branch and bound and cutting planes. Consider the problem formulated as,

$$\begin{aligned} &\min_x f(x) \\ &\text{subject to } x \in C \\ &x_j \in \mathbb{Z}, j \in J \end{aligned}$$

where $f : \mathbb{R}^n \to R$, $C \subseteq \mathbb{R}^n$ are convex and $J \subseteq \{1, \cdots, n\}$

1. Solve the convex relaxation (CR)

$$\begin{aligned} &\min_x f(x) \\ &\text{subject to } x \in C \end{aligned}$$

2. (CR) infeasible $\implies$ (IP) is infeasible.
3. Solution $x^*$ to (CR) is (IP) feasible $\implies$ $x^*$ solution to (IP).
4. Solution $x^*$ to (CR) is not (IP) feasible Choose between two alternatives
    (a) Add cuts and go back to step 1
    (b) Branch and recursively solve subproblems

## 25.6 Integer programming technology

1. State-of-the-art solvers (Gurobi, CPLEX, FICO) rely on extremely efficient implementations of numerical linear algebra, simplex, interior-point, and other algorithms for convex optimization.
2. For mixed integer optimization, most solvers use a clever type of branch and cut algorithm. They rely extensively on convex relaxations. They also take advantage of warm starts as much as possible.
3. Some interesting numbers
    (a) Speedup in algorithms 19902016: over 500,000
    (b) Speedup in hardware 19902016: over 500,000
    (c) Total speedup over 250 billion $= 2.5 \cdot 10^{11}$

## 25.7 Best subset selection

Assume $X = [x^1 \cdots x^p] \in \mathbb{R}^{n\times p}$ and $y \in \mathbb{R}^n$ Best subset selection problem:

$$\begin{aligned} &\min_\beta \frac{1}{2}\|y - X\beta\|_2^2 \\ &\text{subject to } \|\beta\|_0 \le k \end{aligned}$$

Here $\|\beta\|_0 :=$ number of nonzero entries of $\beta$

Let's reformulate it into the integer programming form,

$$\begin{aligned} \min_{\beta,z} \frac{1}{2}\|y - X\beta\|_2^2 \\ \text{subject to } \beta_i \leq M_i \cdot z_i,\ i = 1 \cdots p \\ \sum_{i=1}^{k} z_i \leq k \\ z_i \in \{0,1\},\ i = 1 \cdots p \end{aligned}$$

Here $M_i$ is some a priori known bound on $|\beta_i|$ for $i = 1, \cdots, p$. They can be computed via suitable preprocessing of $X$, $y$.

### 25.7.1 Discrete first-order algorithm

A clever way to get good feasible solutions?

Consider the problem

$$\min_{\beta} g(\beta) \text{ subject to } \|\beta\|_0 \leq k$$

where $g : \mathbb{R}^p \to \mathbb{R}$ is smooth convex and $\nabla g$ is L-Lipschitz.

Best subset selection corresponds to $g(\beta) = \frac{1}{2}\|X\beta - y\|_2^2$

We can observe that, for $u \in \mathbb{R}^p$ the vector

$$H_k(u) = \text{arg min}_{\beta:\|\beta\|_0 \leq k} \|\beta - u\|_2^2$$

is obtained by retaining the $k$ largest entries of $u$. We can do,

1. start with some $\beta(0)$
2. for $i = 0, 1, \cdots$

$$\beta^{(i+1)} = H_k(\beta^{(i)} - \frac{1}{L}\nabla g(\beta^{(i)}))$$

   end for

The above iterates satisfy $\beta(i) \to \bar{\beta}$ where

$$\bar{\beta} = H_k(\bar{\beta} - \frac{1}{L}\nabla g(\bar{\beta}))$$

We therefore achieve some local solution to the above minimization problem.

### 25.7.2 Computational results (Bertsimas et al.)

See figures 25.1, 25.2 and 25.3.

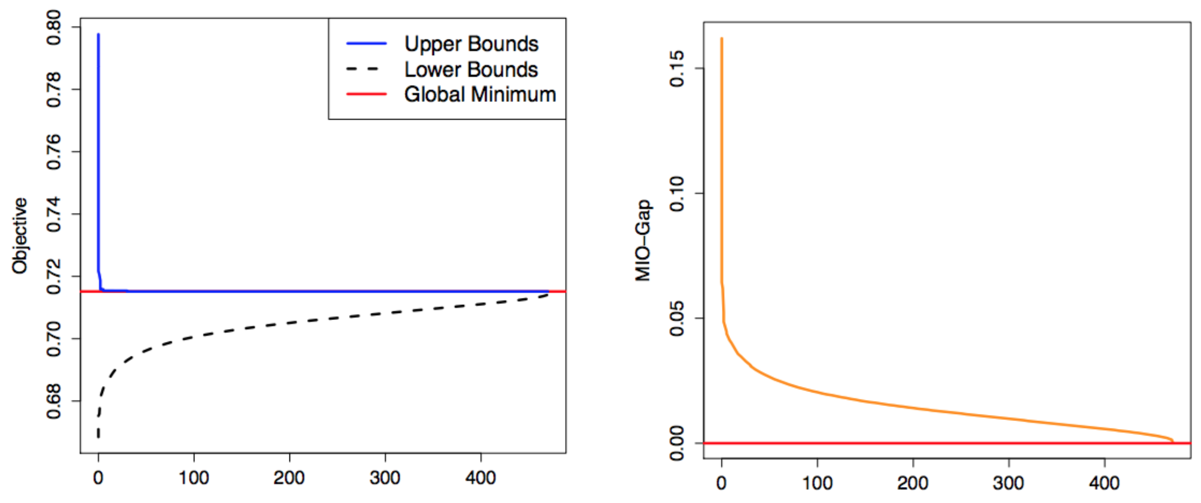

Figure 25.1: Diabetes dataset, $n = 350, p = 64, k = 6$

## 25.8 Least Median of Squares Regression

We are familiar with Least Squares (LS) Regression and Least Absolute Deviation (LAD) Regression which are defined as follows:

$$r_i := y - X\beta \tag{25.1}$$

$$\beta_{LS} := \underset{\beta}{\operatorname{argmin}} \sum_i r_i^2 \tag{25.2}$$

$$\beta_{LAD} := \underset{\beta}{\operatorname{argmin}} \sum_i |r_i| \tag{25.3}$$

While LAD is a less sensitive to outliers as compared to LS, the breakdown point of both LS and LAD estimators is worse than that of Least Median of Squares Regression, which is defined as follows:

$$\beta_{LMS} = \underset{\beta}{\operatorname{argmin}} \left( median|r| \right) \tag{25.4}$$

This is because it is enough to corrupt a single data point to corrupt the LS and LAD estimators, whereas we can corrupt up to half the data without affecting the LMS estimator. However, while we gain in estimator robustness, we lose in tractability. Both LS and LAD solutions can be found easily (analytically for the former and as an LP for the latter). Finding the optimal LMS estimator on the other hand is an NP hard problem.

LMS Regression can be generalized to Least Quantile (LQ) Regression as follows:

$$\beta_{LQS} := \underset{\beta}{\operatorname{argmin}} |r_{(q)}| \tag{25.5}$$

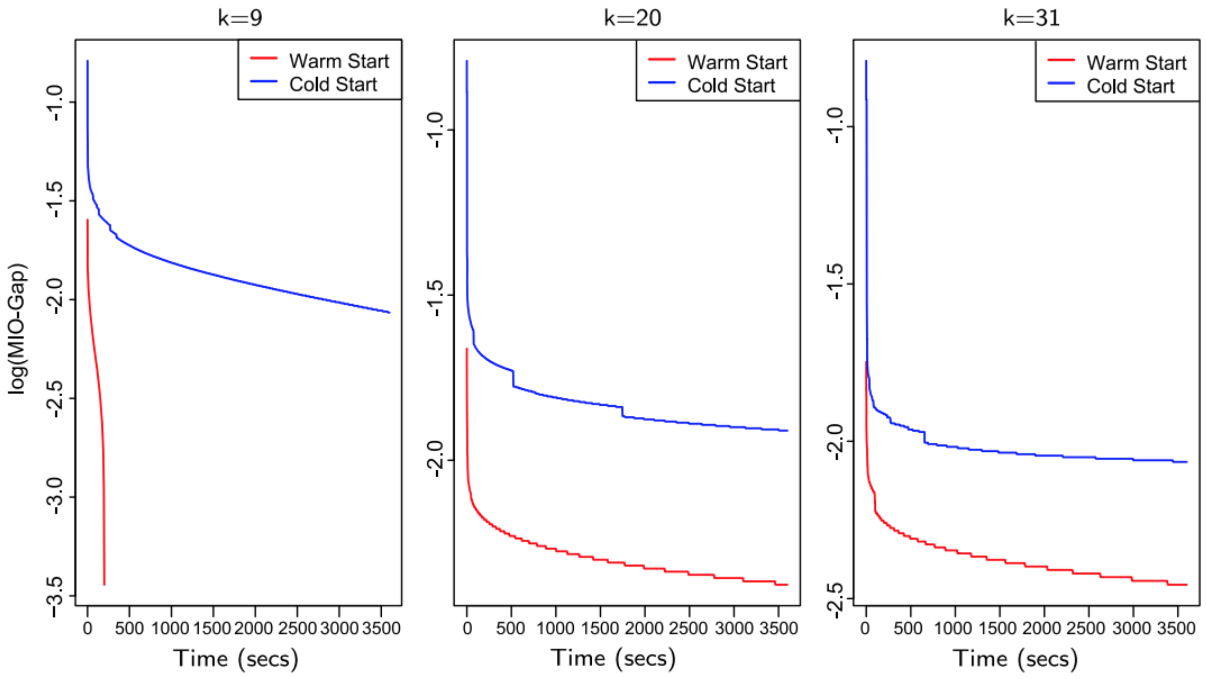

Figure 25.2: Cold vs Warm Starts

where $r_{(q)}$ is the $q^{th}$ ordered residual where:

$$r_{(1)} \leq r_{(2)} \leq ... \leq r_{(n)} \tag{25.6}$$

LMS is a special case of LQ when q $= \frac{n}{2}$. LQ Regression can be cast as a Mixed Integer Program (MIP) by using binary and auxiliary variables to encode the conditions

$$|r_i| \leq |r_{(q)}| \text{ or } |r_i| \geq |r_{(q)}| \text{ for each entry i of r}$$

Let $\gamma = r_{(q)}$. LQ regression can be cast as an MIP as follows:

$$
\begin{aligned}
&\min_{\beta,\mu^{(1)},\mu^{(2)},z,\gamma} \gamma && (25.7)\\
\text{subject that: } &\gamma \le |r_i| + \mu^{(1)},\ \text{i} = \{\ 1, \dots\ , \text{n}\} && (25.8)\\
&\gamma \ge |r_i| + \mu^{(2)},\ \text{i} = \{\ 1, \dots\ , \text{n}\} && (25.9)\\
&\mu^{(1)} \le M z_i,\ \text{i} = \{\ 1, \dots\ , \text{n}\} && (25.10)\\
&\mu^{(2)} \le M(1 - z_i),\ \text{i} = \{\ 1, \dots\ , \text{n}\} && (25.11)\\
&\sum_{i=1}^{p} z_i = q && (25.12)\\
&\mu_i^{(1)}\mu_i^{(2)} \ge 0,\ \text{i} = \{\ 1, \dots\ , \text{n}\} && (25.13)\\
&z_i \in \{0, 1\} && (25.14)
\end{aligned}
$$

Note that the first constraint effectively lower bound an absolute value which is a non-convex constraint. This could be converted into a convex constraint by introduction of additional variables.

Having converted LQ regression into an MIP, a first order algorithm for solving it can be obtained. First, observe that:

$$|r_{(q)}| = |y_{(q)} - x_{(q)}^T\beta| = H_q(\beta) - H_{q+1}(\beta) \tag{25.15}$$

where

$$
\begin{aligned}
H_q(\beta) = \sum_{i=q}^{n} |y_{(i)} - x_{(i)}^T\beta| = \qquad & max_w \sum_{i=1}^{n} w_i |y_{(i)} - x_{(i)}^T\beta| \\
\text{subject to: } & \sum_{i=1}^{n} w_i = q \\
& 0 \le w_i \le 1,\ i = 1, \dots, n
\end{aligned}
$$

Note that function $H_q(\beta)$ is convex. Thus, we can use the subgradient algorithm to obtain the local minimum of $H_q(\beta) - H_{q+1}(\beta)$. The computational results of this are shown in figures 25.4 and 25.5.

## References

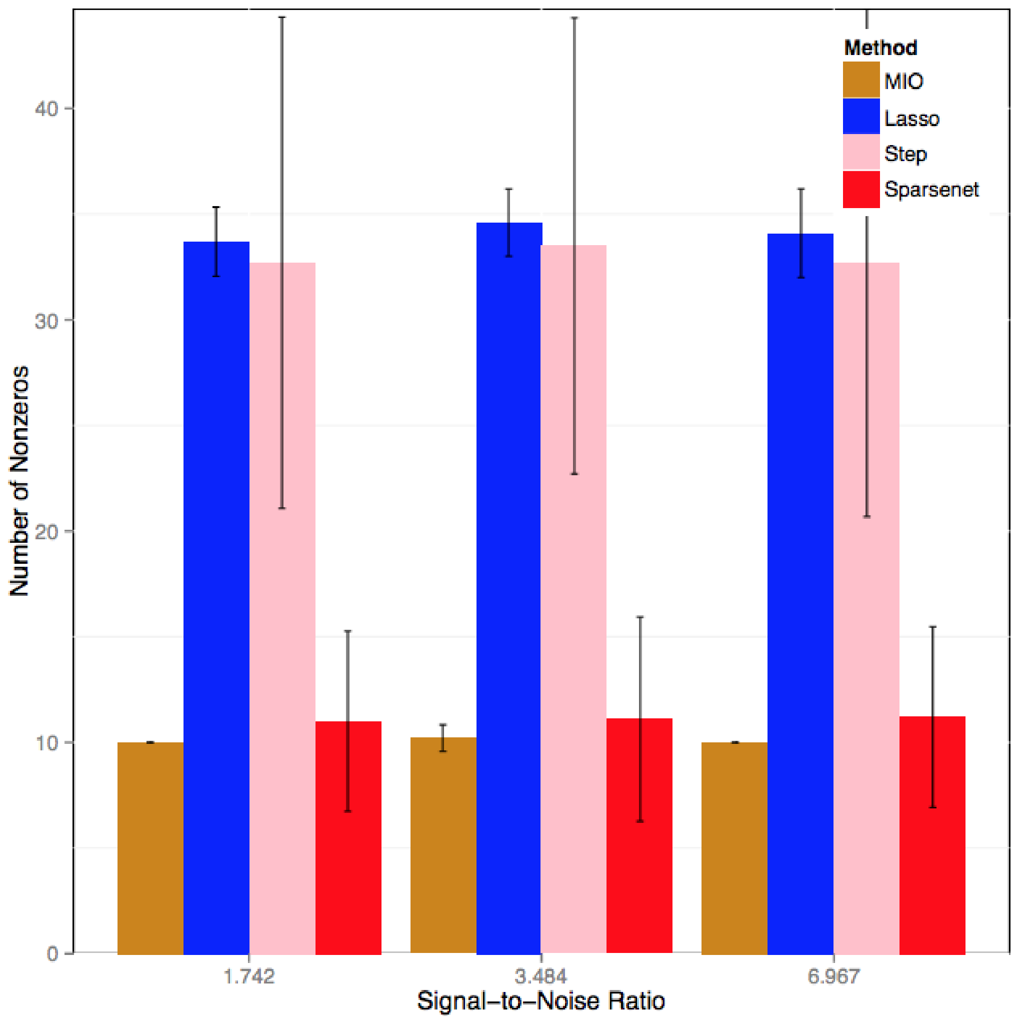

Figure 25.3: Sparsity detection (synthetic datasets)

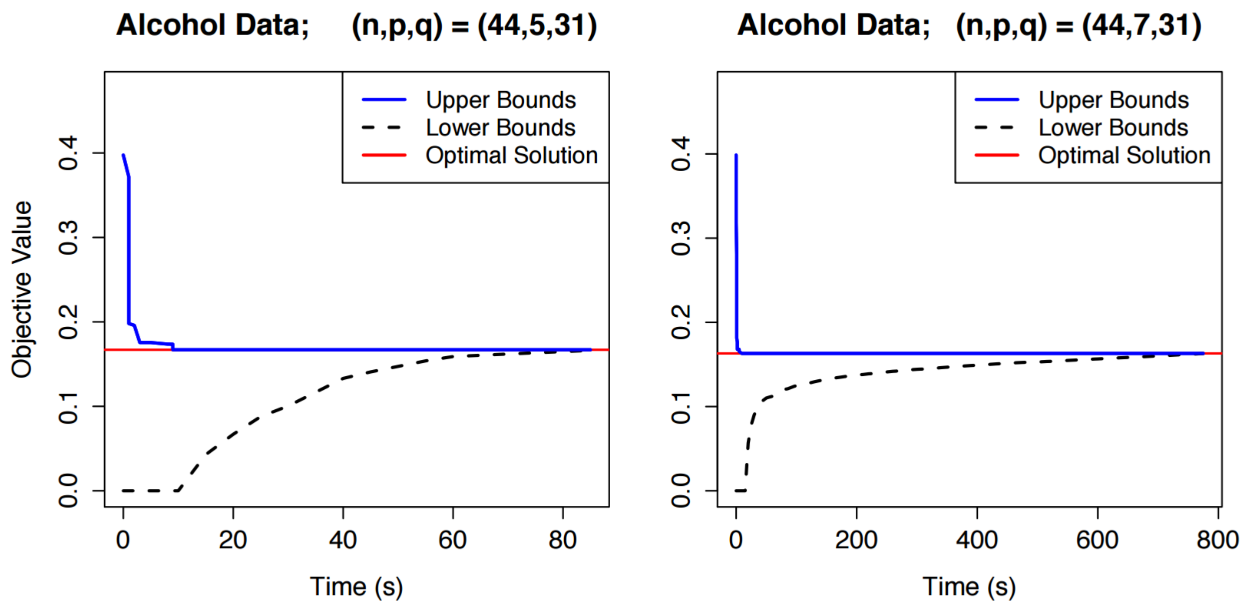

Figure 25.4: Alcohol dataset, (n,p,q) = (44,5,31)

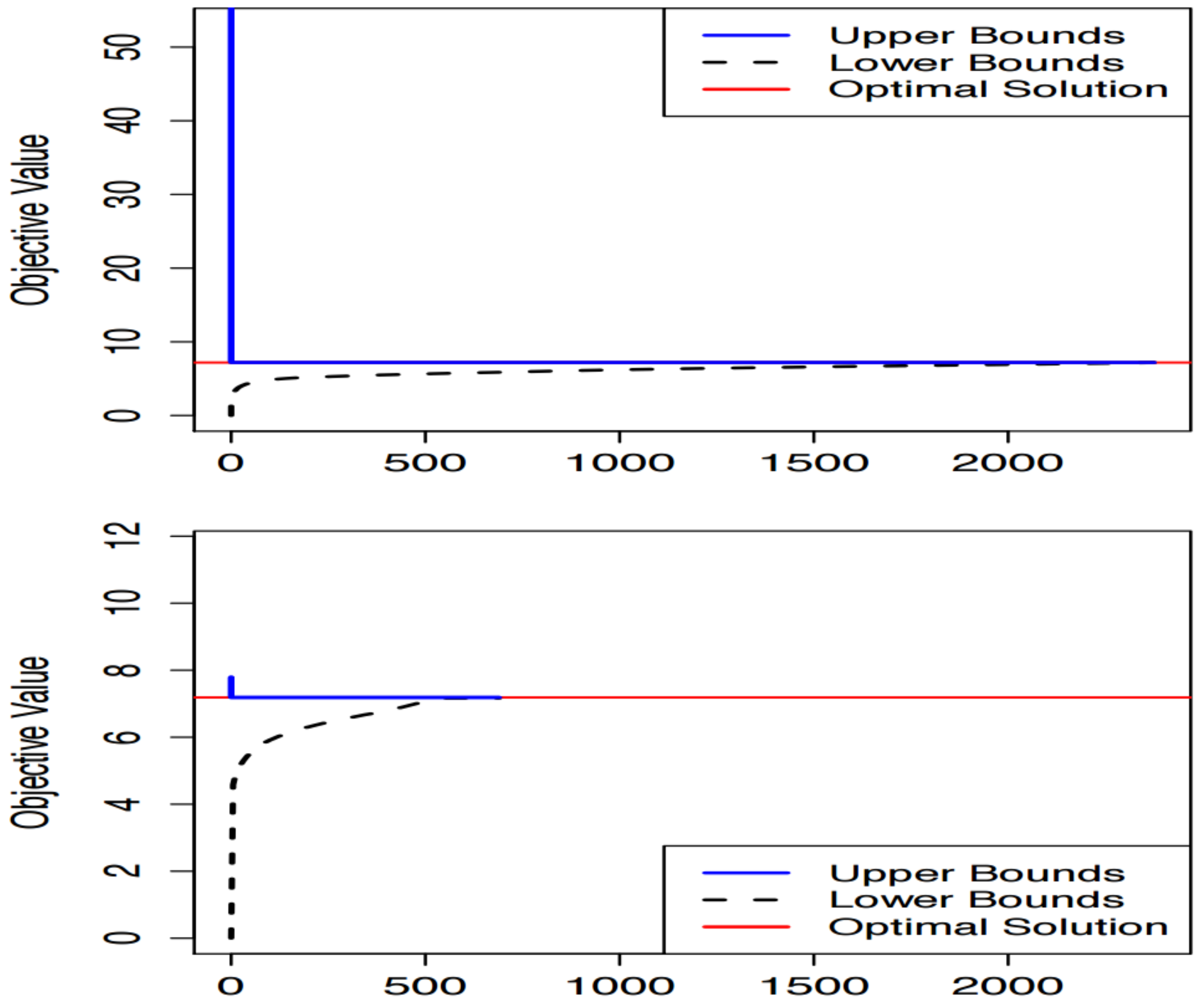

Figure 25.5: Cold vs Warm Starts