University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations

1999

Redundancy in nonlinear systems: A set covering approach.

Jinghua. Feng
University of Windsor

Redundancy in Nonlinear Systems: A Set Covering Approach

by

JINGHUA FENG

A Thesis
Submitted to the College of Graduate Studies and Research
through the Department of Economics, Mathematics and Statistics
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

1999

0-612-52546-5

Abstract

In this thesis we present Boneh's Set Covering (SC) approach to the redundancy detection problem, and we show that his approach is also applicable to the related problem of finding a Prime Representation (PR), an Irreducible Infeasible System (IIS) or a Minimal Infeasible System (MIS).

In order to generate the SC matrix E, we need a probabilistic method for sampling points in $\mathbf{R}^n$. Consequently we can assign a detection probability to each row of E, and we show that if a row of E has a zero detection probability, then it must correspond to what we call a local quasi-minimizer. We show that convex systems have no such local quasi-minimizers.

To my parents and my wife

Acknowledgements

First and foremost, I must thank my supervisor, Dr. R.J. Caron, who has been giving me encouragement and support from the very beginning of my study in this department and throughout the development of this thesis. Also, thanks for his guidance, constructive criticism, and much help and energy devoted to correct my linguistic oddities.

I would express my gratitude to Dr. T. Traynor for his wholehearted support and assistance in the preparation of this thesis, and for serving as my co-supervisor.

I would also like to thank Dr. Yash P. Aneja for serving as member of my committee.

Also, I would like to thank Dr. Zhiguo Hu for answering my many questions.

Finally, I wish to express my utmost gratitude to my parents and my wife. Without their understanding, encouragement, and love, it would have been impossible for me to complete this work.

Contents

List of Definitions

List of Figures

1 Introduction

1.1 Introduction

In this thesis, we are concerned with systems of constraints that arise in mathematical programming problems. Mathematical programs are important because of their widespread application to decision problems in business, industry, academic research and government. A mathematical program is the problem of determining the value of a set of variables that will satisfy a set of restrictions and that will also optimize an objective which depends on those variables. Such problems can be extremely large with hundreds of thousands of constraints and variables.

We are interested in the set of restrictions, that is, in the system of mathematical inequalities, which model those restrictions. In any large system of constraints, it is quite likely that as many as 50% of the constraints are redundant [4]; that is, they can be eliminated without effecting the restrictions on the variables. This redundancy problem, particularly for linear constraints, has been studied by many authors. In this thesis, we will consider a particular probabilistic technique introduced by Boneh that can be used to identify redundancy for very general constraints. We will highlight the desirable features and a shortcoming of Boneh's approach, and we will provide results that attempt to explain that shortcoming.

1.2 Definitions and notation

We are concerned with the set of constraint functions

$$C(I) := \{ g_i(x) \mid i \in I \},$$

where I is an index set. We assume that the constraint functions are continuous functions from $\mathbf{R}^n$ to $\mathbf{R}$. With each constraint function, we have the constraint feasibility region

$$R_i = \{ x \in \mathbf{R}^n \mid g_i(x) \leq 0 \}$$

and the constraint surface set

$$S_i = \{ x \in \mathbf{R}^n \mid g_i(x) = 0 \}.$$

We will refer to S_i as the "surface" of constraint i. (The use of the word "surface" is not meant to imply anything about the topological properties of S_i).

The feasible region for the constraint set $C(I)$ is given by

$$R(I) = \{ x \in \mathbf{R}^n \mid g_i(x) \leq 0, \ \forall i \in I \} = \bigcap_{i \in I} R_i.$$

We note that $R(I)$ may, in fact, be empty; that is, we may have $R(I) = \emptyset$. We adopt the convention that $R(\emptyset) = \mathbf{R}^n$.

Also, we use the symbols $\text{int}(A)$ and $\partial(A)$ to denote the interior and boundary, respectively, of the set A in $\mathbf{R}^n$. The $\text{int}(A)$ is given by

$$\text{int}(A) = \{ x \in A \mid \exists \epsilon > 0 \text{ with } \mathbf{B}(x, \epsilon) \subseteq A \}$$

where $\mathbf{B}(x, \epsilon)$ is the open ball centered at x with radius ϵ. Note that $\partial(A) = \text{cl}(A) \setminus \text{int}(A)$.

Definition 1.1 We say the set A is *full dimensional* if $\text{int}(A) \neq \emptyset$.

In what remains of this section, we will provide definitions and examples to illustrate concepts related to redundancy. We note that the definitions do not presuppose that $R(I) \neq \emptyset$. It is evident that $I \supseteq \hat{I}$ implies $R(I) \subseteq R(\hat{I})$; it is important to know at what stage such an inclusion becomes proper.

Definition 1.2 The k-th constraint is *necessary* in $C(I)$ if $R(I \setminus \{k\}) \neq R(I)$ and is *redundant* in $C(I)$ if $R(I \setminus \{k\}) = R(I)$.

Definition 1.3 $C(\hat{I})$ is *a set of duplicate constraints* if for each pair $i, j \in \hat{I}$, $R_i = R_j$.

Example 1.1 Consider the set of seven linear constraints given by

$$\begin{aligned} g_1(x) &= -x_1 + x_2 \\ g_2(x) &= x_1 + x_2 - 3 \\ g_3(x) &= x_1 - 2 \\ g_4(x) &= -x_2 \\ g_5(x) &= -x_1 + x_2 - 1 \\ g_6(x) &= x_1 + x_2 \\ g_7(x) &= 2x_1 + 2x_2 - 6 \end{aligned}$$

that are graphed in Figure 1.

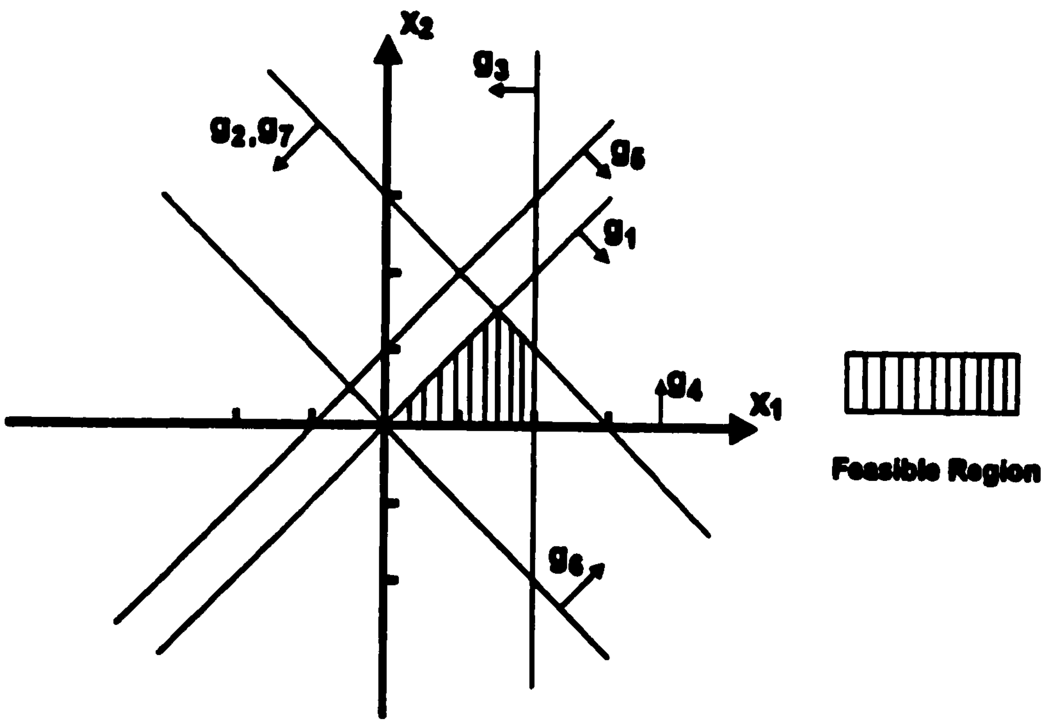

Figure 1: Redundant, necessary and duplicate constraints

In Figure 1, the feasible region $R(I)$ is shaded, the surface sets S_i are labeled as g_i, and the arrows indicate the constraint feasibility regions R_i. From the diagram, it is clear that constraints 1, 3 and 4 are necessary in $C(I)$ and constraints 2, 5, 6 and 7 are redundant in $C(I)$. Note that the constraint set $C(\{2, 7\})$ is a set of duplicate constraints. Consequently, constraint 2 is redundant in $C(I)$. However, it is necessary in $C(I \setminus \{7\})$. Likewise, constraint 7 is redundant in $C(I)$ but is necessary in $C(I \setminus \{2\})$. This illustrates the fact that whether or not a constraint is redundant is relative to the constraint set in question.

□

Definition 1.4 The i-th constraint is *an implicit equality* in $C(I)$ if

$$R(\hat{I}) \subseteq S_i \text{ and } R(\hat{I}) \neq \emptyset,$$

for some $\hat{I} \subseteq I$.

At first, this definition might seem a little unusual. However, if $R(I) \neq \emptyset$, then we see that, as we expect, a constraint "$g_i(x) \leq 0$" is an implicit equality if $g_i(x) = 0$ for all $x \in R(I)$. We have given a generalization of the usual definition to account for the possibility that $R(I) = \emptyset$, that is, that the system is infeasible.

Example 1.2 Consider the set of four linear constraints given by

$$\begin{aligned} g_1(x) &= x_1 - x_2 \\ g_2(x) &= x_1 + x_2 \\ g_3(x) &= -x_1 \\ g_4(x) &= -x_1 + 2 \end{aligned}$$

that are graphed in Figure 2.

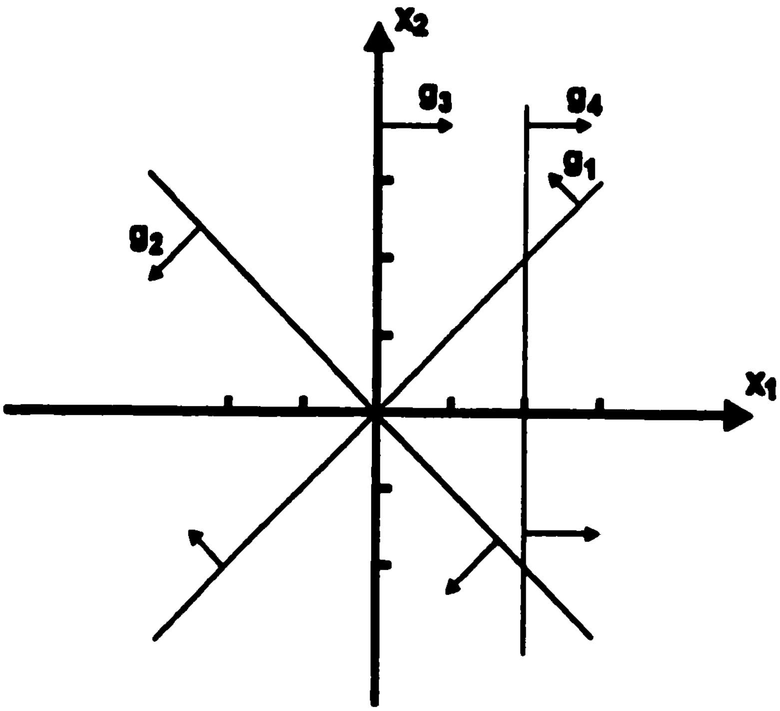

Figure 2: Implicit equality constraints

In Figure 2, we note that $R(I) = \emptyset$. Constraints 1, 2 and 3 are implicit equality constraints in $C(I)$ since

$$R(\hat{I}) \subset S_i,\ i = 1,2,3, \text{ and } R(\hat{I}) = \{\,(0,0)\,\} \neq \emptyset,$$

where $\hat{I} = \{ 1, 2, 3 \}$.

□

In this thesis, we assume that there are no implicit inequalities in $C(I)$.

The next two definitions deal with reductions of the constraint set that can be achieved by the removal of redundancy. The symbol " $\subset$ " is used to denote proper inclusion.

Definition 1.5 $C(\hat{I})$ is a *reduction* of $C(I)$ if $R(\hat{I}) = R(I)$ and $\hat{I} \subset I$.

Definition 1.6 $C(\hat{I})$ is *irreducible* if $R(\hat{I}) \neq R(\bar{I})$ for every $\bar{I} \subset \hat{I}$. If also $R(\hat{I}) = \emptyset$, then $C(\hat{I})$ is an *Irreducible Infeasible System* (IIS) [9]; and if $R(\hat{I}) \neq \emptyset$, then $C(\hat{I})$ is a *Prime Presentation* (PR) of $R(\hat{I})$ [5].

Example 1.3 Consider the system of Example 1.1. The index set $I = \{ 1, 2, 3, 4, 5, 6, 7 \}$. From Figure 1, we see that $R(\hat{I}) = R(I)$, where $\hat{I} = \{ 1, 2, 3, 4 \}$, so $C(\hat{I})$ is a reduction of $C(I)$. It should also be clear that $C(\hat{I})$ is irreducible. Similarly, for $\bar{I} = \{ 1, 3, 4, 7 \}$, $C(\bar{I})$ is a reduction of $C(I)$ and is irreducible.

□

Example 1.4 Consider the system of Example 1.2. We have $I = \{1, 2, 3, 4\}$. From Figure 2, we see that $R(\hat{I}) = R(I)$, where $\hat{I} = \{1, 2, 4\}$. Hence, $C(\hat{I})$

is a reduction of $C(I)$, and since $R(\hat{I}) = \emptyset$, $C(\hat{I})$ is an Irreducible Infeasible System (IIS).

□

Definition 1.7 Let $\hat{I} \subseteq I$. We say that $C(\hat{I})$ is an *infeasibility set* with respect to $C(I)$ if $R(I) = \emptyset$ and $R(I \setminus \hat{I}) \neq \emptyset$. A *minimal infeasibility set* (MIS) is one which has minimum cardinality [8].

Example 1.5 In Example 1.2, we have $I = \{1, 2, 3, 4\}$ and $R(I) = \emptyset$. $C(\{1\})$, $C(\{2\})$ and $C(\{4\})$ are the MIS's, and any constraint set which contains one of these is therefore an infeasibility set.

□

1.3 Outline of thesis

This thesis is mainly concerned with the application of a particular probabilistic method, introduced by Boneh, to various redundancy problems for general systems of constraints, such as finding a PR, an IIS, or a MIS, that is, a Prime Representation, an Irreducible Infeasible System or a Minimal Infeasibility Set.

In Chapter 2 we give descriptions of the standard deterministic and probabilistic methods for redundancy. We then introduce Boneh's method, which, as we will see, is based on an equivalence between the redundancy problem and a set covering feasibility problem.

In order to use Boneh's set covering techniques, a method of sampling points in $\mathbf{R}^n$ is needed to generate the rows of the Set Covering (SC) feasibility matrix. An example is used to indicate that, for some standard sampling procedures, the probability of obtaining all rows of the SC matrix is zero. This phenomenon is explored in Chapter 3 which is the main contribution of the thesis. In Chapter 4, we show that the phenomenon does not occur in convex problems. Chapter 5 summarizes the results of this thesis and discusses future research directions.

2 Historical Perspective

2.1 Introduction

The earliest paper devoted entirely to redundancy was given by Boot [6]. In that paper, he introduced what is now known as the "turnover lemma". This lemma can be viewed as the forerunner of the deterministic methods, which are discussed in section 2.2.

As an alternative to the deterministic methods, there is a class of probabilistic methods known as "hit-and-run" methods. These are described in section 2.3.

The methods of 2.2 have the limitation that they are mainly restricted to linear constraints, and the methods in 2.3 require a nonempty feasible region as well as certain regularity conditions on the constraint functions. This prompted Boneh to propose his set covering method which is described in 2.4.

2.2 Deterministic methods

The following theorem is the foundation of the deterministic methods.

Theorem 2.1 (Boot's Turnover Lemma [6]) Fix an index $k \in I$. The set $\{x \in R(I \setminus \{k\}) \mid g_k(x) > 0\} \neq \emptyset$ if and only if the k-th constraint is necessary in $C(I)$. Equivalently, $\{ x \in R(I \setminus \{k\} \mid g_k(x) > 0 \} = \emptyset$ iff the k-th constraint is redundant in $C(I)$.

Proof: If $\{ x \in R(I \setminus \{k\}) \mid g_k(x) > 0 \} \neq \emptyset$, then there exists a $\hat{x}$ such that $g_i(\hat{x}) \leq 0,\ i \in I \setminus \{k\}$ and $g_k(\hat{x}) > 0$. So, $\hat{x} \in R(I \setminus \{k\})$ and $\hat{x} \notin R(I)$, i.e., $R(I \setminus \{k\}) \neq R(I)$. From Definition 1.2, the k-th constraint is necessary.

Conversely, if the k-th constraint is necessary in $C(I)$ then, by Definition 1.2, $R(I \setminus \{k\}) \neq R(I)$. Since $R(I) \subseteq R(I \setminus \{k\})$, it then follows that there exists an $\hat{x}$ such that $\hat{x} \in R(I \setminus \{k\})$, and $\hat{x} \notin R(I)$, thus, $\hat{x} \notin R_k$ and $\{ x \in R(I \setminus \{k\}) \mid g_k(x) > 0 \} \neq \emptyset$.

□

Theorem 2.2, below, shows how the feasibility required by Theorem 2.1 can be converted to an optimization problem if for each $k \in I$, the maximum of $\{ g_k(x) \mid x \in R(I) \}$ exists when $R(I) \neq \emptyset$. Situations under which the required maximum exists include (1) $R(I)$ is bounded (hence compact) or (2) all constraint functions are linear.

Theorem 2.2 Under the above assumption, the k-th constraint is redundant in $C(I)$ if and only if either $R(I \setminus \{k\}) = R(I) = \emptyset$ or $R(I) \neq \emptyset$

and

$$\text{MP}_\text{k}: \qquad \text{maximize}\ \{\, g_k(x) \mid x \in R(I \setminus \{k\})\,\}$$

has an optimal solution, say $\hat{x}$, satisfying $g_k(\hat{x}) \leq 0$.

Proof: For the if part, if $R(I) = R(I \setminus \{k\}) = \emptyset$, then the k-th constraint is redundant by Definition 1.2.

If $R(I) \neq \emptyset$ and MP_k has the optimal solution $\hat{x}$ with $g_k(\hat{x}) \leq 0$, then for all $x \in R(I \setminus \{k\})$ we have $g_k(x) \leq g_k(\hat{x}) \leq 0$, thus $R(I \setminus \{k\}) \subseteq R(I)$. Since $R(I \setminus \{k\}) \supseteq R(I)$, we have $R(I \setminus \{k\}) = R(I)$ and k-th constraint is redundant by Definition 1.2.

For the only if part, the k-th constraint is redundant. If $R(I) = \emptyset$, then, $R(I \setminus \{k\}) = R(I) = \emptyset$.

If $R(I) \neq \emptyset$. Since g_k is redundant, $R(I \setminus \{k\}) = R(I)$. Since max $\{\, g_k(x) \mid x \in R(I)\,\}$ is assumed to exist, let $\hat{x}$ be an optimal solution. Then $g_k(\hat{x}) = \max\ \{\, g_k(x) \mid x \in R(I \setminus \{k\})\,\} = \max\ \{\, g_k(x) \mid x \in R(I)\,\} \leq 0$, so, $g_k(\hat{x}) \leq 0$.

□

Based upon this theorem, we can easily devise a simplistic algorithm for the classification of the constraints in $C(I)$ as either redundant or necessary.

Notice that the algorithm deletes redundant constraints as they are found. Since redundancy is relative to the constraint set, the resulting irreducible set $C(\hat{I})$ depends on the ordering of the constraints in the original constraint set.

A deletion-filtering algorithm

Step 1: Input I and $C(I)$, set $\hat{I} = I$.

Step 2: For each $k \in I$ do:

- 2.1: Solve MP_k : maximize $\{ g_k(x) \mid x \in R(\hat{I} \setminus \{k\}) \}$.
- 2.2: If MP_k is infeasible then $R(\hat{I} \setminus \{k\}) = R(\hat{I}) = \emptyset$ and the k-th constraint is redundant. If MP_k has an optimal solution $\hat{x}$ with $g_k(\hat{x}) \leq 0$, then the k-th constraint is also redundant. Otherwise, the k-th constraint is necessary.
- 2.3: If the k-th constraint is redundant, replace $\hat{I}$ with $\hat{I} \setminus \{k\}$ and continue with the next index k.

Step 3: Output $C(\hat{I})$, an irreducible reduction of $C(I)$.

We first note that the algorithm can be applied to feasible $R(I) \neq \emptyset$ and infeasible $R(I) = \emptyset$ problems. We mention this because most researchers see the two cases as distinct with papers either discussing redundancy, when $R(I) \neq \emptyset$, or infeasibility, when $R(I) = \emptyset$; this will be more apparent in our discussion of the linear case.

Let's suppose that $C(I)$ is a set of linear constraints. If $R(I) \neq \emptyset$, then the algorithm produces a Prime Representation $C(\hat{I})$ of $R(I)$ (Definition 1.6). Indeed, if there are no implicit equalities, then $C(\hat{I})$ is a "minimal representation" of $R(I)$ [18]. For feasible systems, the deletion-filtering algorithm is, in fact, the basis for all the simplex based methods in Karwan [13] and the method in Caron et al. [7]. If $R(I) = \emptyset$, then the deletion-filtering algorithm is the foundation of Chinneck's methods [9] for the detection of IIS's. Indeed, it is from his paper that we have taken the term "deletion-filtering".

While the deletion-filtering algorithm provides a useful framework from which to view the redundancy classification problem, its usefulness is really restricted to linear constraints. Even then, its effectiveness depends upon clever implementation and enhancements. Because of these shortcomings, researchers were drawn to the probabilistic methods.

2.3 Probabilistic hit-and-run methods

We present the hypersphere direction (HD) method as representative of the class of hit-and-run probabilistic methods [2].

An iteration of HD starts with a feasible interior point, say x_j (refer to Figure 3). In the hit step, we first generate a random direction vector s_j.

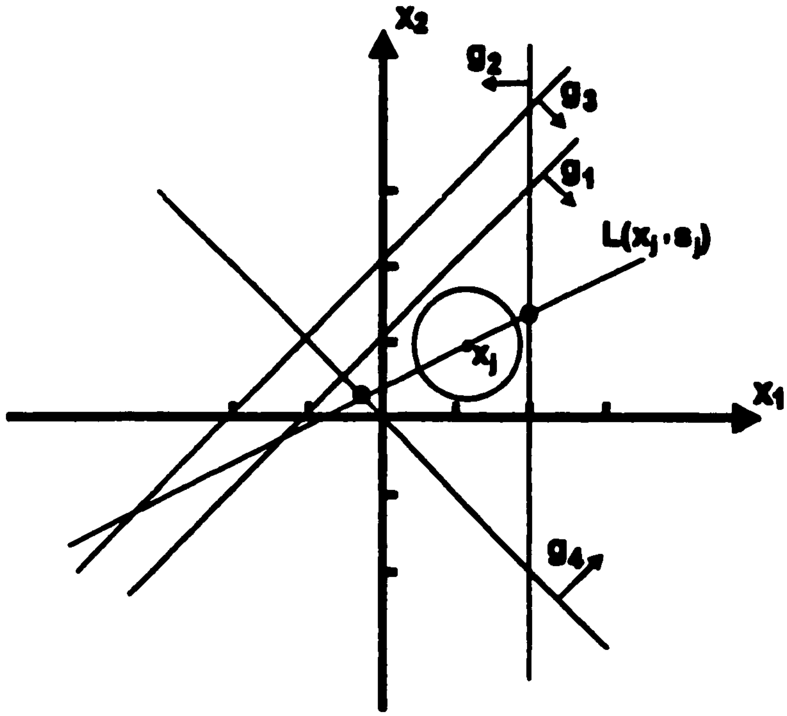

Figure 3: Iterations of HD algorithm

This is done by selecting a point uniformly distributed over the surface of the n dimensional hypersphere H $= \{x \in \mathbf{R}^n \mid ||x|| = 1\}$ (To obtain a search direction $s_j \sim U(H)$, we generate $z_i \sim N(0,1)$, and set $s_j = z/||z||$, where $z = (z_1, \ldots, z_n)^T$. The uniform distribution on H is a consequence of the radial symmetry of the multivariate normal distribution [14]). Together, x_j and s_j define a line $L(x_j, s_j) = \{x_j + \sigma s_j \mid \sigma \in \mathbf{R}\}$ in $\mathbf{R}^n$ which passes through $R(I)$. We then determine the feasible segment of the line by calculating the intersection points of the line with the constraint surfaces S_i. The inequalities whose surfaces are hit by the end points of the feasible line segment are said to have been detected, and are classified as necessary. In the run step, we generate a new interior point x_{j+1} from a uniform distribution over the feasible line segment. These steps are repeated until a stopping rule is satisfied and, upon termination, all constraints that have not been detected are classified, possibly with error, as redundant.

A HD hit-and-run algorithm

Given	A constraint set $C(I)$ with $x_0 \in \text{int}(R(I))$, $R(I)$ bounded.
Initialization	Set $j = 0$, $\hat{I} = I$.
Repeat	
(Search Direction)	Generate $z_i \sim N(0,1)$, and set $s_j = z/\|\|z\|\|$, where $z = (z_1, \ldots, z_n)^T$.
(Hit)	For each $i \in I$ determine $\sigma_i^- = \max\{\sigma \mid g_i(x_j + \sigma s_j) = 0,\ \sigma < 0\}$ and $\sigma_i^+ = \min\{\sigma \mid g_i(x_j + \sigma s_j) = 0,\ \sigma > 0\}$. Determine l and u such that $\sigma_l = \max\{\sigma_i^- \mid i \in I\}$ and $\sigma_u = \min\{\sigma_i^+ \mid i \in I\}$. The two hit points are $x_j + \sigma_l s_j$ and $x_j + \sigma_u s_j$. If l and u are unique, then constraints l and u are necessary. Set $\hat{I} = \hat{I} \setminus \{l, u\}$.
(Run)	Let $u \sim U(0,1)$, and $x_{j+1} = x_j + (\sigma_l + u(\sigma_u - \sigma_l))s_j$.
Until	If the stopping rule holds, output the redundant constraint set $\hat{I}$ and the necessary constraint set $I \setminus \hat{I}$, otherwise set $j = j + 1$.

Note that the HD algorithm requires $R(I)$ to be nonempty, full dimensional, and bounded. The next theorem justifies the algorithm.

Theorem 2.3 [2] In the hit-and-run algorithm above, if σ_l (σ_u) is defined by a unique index $l(u)$, then, the constraint l (u) corresponding with σ_l (σ_u) is necessary.

Proof: We prove the σ_l case, the other is similar. From the uniqueness of the index l, we have $g_l(x_j+\sigma_l s_j) = 0$ and $g_i(x_j+\sigma_l s_j) < 0$, for $i \in I\backslash\{l\}$. Then for sufficiently small $\epsilon > 0$, $g_l(x_j + (\sigma_l - \epsilon)s_j) > 0$ and $g_i(x_j + (\sigma - \epsilon)s_j) < 0$, for $i \in I \setminus \{l\}$, i.e. $(x_j + (\sigma_l - \epsilon)s_j) \in R(I \setminus \{l\})$, but $(x_j + (\sigma_l - \epsilon)s_j) \notin R(I)$. Hence $R(I) \neq R(I \setminus \{l\})$, and constraint l is necessary.

□

Suppose that σ_l is not defined by a unique index l. Then $I(\sigma_l) = \{ i \in I \mid g_i(x_j + \sigma_l s_j) = 0 \}$ is such that $|I(\sigma_l)| > 1$. There are two cases. Either there exists an $\epsilon > 0$ such that $\{ x \in \mathbf{B}(x_j + \sigma_l s_j, \epsilon) \mid g_i(x) = 0,\ i \in I(\sigma_l) \} = \{ x_j + \sigma s_j \}$ or $\{ x \in \mathbf{B}(x_j + \sigma_l s_j, \epsilon) \mid g_i(x) = 0, i \in I(\sigma_l) \} \supset \{ x_j + \sigma s_j \}$. The former possibility can only happen with probability 0 and is therefore disregarded. The latter possibility is assumed not to occur. If the constraints are convex quadratic or linear, then the latter case only occurs if there are duplicate constraints [16]. These can be detected and removed by a modification of the HD algorithm. This is because if $\sigma_i = \sigma_k$, then, with probability one, constraints i and k are duplicates and one of these can be removed.

Clearly in a hit-and-run algorithm, the cost of an iteration is dominated by the computations of the intersection of the line segment with the constraint surface sets. For linear constraints, we have, say, that $g_i(x) = a_i^T x - b_i$, so that

$$\sigma_i = \frac{b_i - a_i^T x_j}{a_i^T s_j},$$

which is the usual linear programming step size. For quadratic constraints, $g_i(x) = a_i^T x + \frac{1}{2} x^T B_i x - b_i$, so that, let $\Delta = ((a_i + B_i x_j)^T s_j)^2 - 2(a_i^T x_j + \frac{1}{2} x_j^T B_i x_j - b_i) s_j^T B_i s_j$,

$$\sigma_i = \frac{-(a_i + B_i x_j)^T s_j \pm \sqrt{\Delta}}{s_j^T B_i s_j}.$$

In other situations one may use other techniques, for example, Newton's method, to approximate the σ_i. Boneh [3] suggested that it is desirable (though not necessary) that as many constraints as possible are invertible. A constraint is said to be invertible if there exists either an explicit closed form solution for the evaluation of its intersection points with an arbitrary straight line or some fast-converging algorithm for finding such points.

Shortcomings of the HD method include the need for a feasible point, the need to solve for the intersection points, and, the constraint function should satisfy the property that, if $g_i(\hat{x}) = g_j(\hat{x}) = 0$, then either for some $\epsilon > 0$,

$$\{ S_i \cap \mathbf{B}(\hat{x}, \epsilon) \} = \{ S_j \cap \mathbf{B}(\hat{x}, \epsilon) \} = \{\hat{x}\},$$

or

$$S_i = S_j.$$

These are overcome by the SC approach.

2.4 The set covering approach

We begin with the following theorem, upon which the set covering approach is based. It shows that at least one of the constraints violated by an infeasible point $\hat{x} \in \mathbf{R}^n$ must be necessary in a reduction of the constraint set $C(I)$.

Theorem 2.4 Let $\bar{I} \subset I$. Then $C(\bar{I})$ is a reduction of $C(I)$ if and only if whenever x is an infeasible point, there exists $i \in \bar{I}$ with $g_i(x) > 0$.

Proof: Since we always have $R(\bar{I}) \supseteq R(I)$, $C(\bar{I})$ is a reduction of $C(I)$ if and only if $R(\bar{I}) \subseteq R(I)$. Considering complements, this holds if and only if each x which does not belong to $R(I)$, does not belong to $R(\bar{I})$. But this means whenever x is an infeasible point, there exists $i \in \bar{I}$ with $g_i(x) > 0$.

The above can be considered a result about set covering. Let $A_i = \{ x \mid g_i(x) > 0 \} = R_i^c$. Theorem 2.4 is saying $C(\bar{I})$ is a reduction of $C(I)$ if and only if $\{ A_i \mid i \in \bar{I} \}$ covers the set $R(I)^c$ of infeasible points:

$$\bigcup_{i \in \bar{I}} A_i \supset R(I)^c.$$

□

Now, let's refer to $x \in \mathbf{R}^n$ as an observation. Corresponding to this observation is the binary word

$$e(x) = (e_1(x), e_2(x), \ldots, e_m(x)),$$

where

$$e_i(x) = \begin{cases} 1 & \text{if } g_i(x) > 0 \\ 0 & \text{if } g_i(x) \leq 0 \end{cases}$$

for $i = 1, 2, \ldots, m$. We also say that for the given constraint set, $e(x)$ is observable.

Now, suppose that $C(\bar{I})$ is a reduction of $C(I)$. Corresponding to this reduction, we can define the binary word $y = (y_1, \ldots, y_m)^T$ where

$$y_i = \begin{cases} 1 & \text{if } i \in \bar{I} \\ 0 & \text{if } i \notin \bar{I} \end{cases}$$

Then, Theorem 2.4 states that if x is an infeasible point,

$$e(x)y \geq 1.$$

Since both $e(x)$ and y are binary words, this can be understood as a constraint for the set covering (SC) problem [10]. Indeed, if we let E be SC matrix containing all possible distinct observations corresponding with infeasible points, then there is a one-to-one correspondence between all reductions and all solutions to the SC constraints

$$\begin{array}{c} Ey \geq \bar{1} \\ y \in \{0,1\}^m \end{array} \tag{1}$$

where $\bar{1} = (1, 1, \ldots, 1)^T$. This is formalized in the following theorem.

Assume $R(I) \neq \mathbf{R}^n$, that is, there exists at least one infeasible point. Also, we let $\bar{0} = (0, 0, \ldots, 0)$.

Theorem 2.5 Let y be a binary word and $\hat{I} = \{\, i \mid y_i = 1 \,\}$. Then y is a feasible solution to (1) if and only if $C(\hat{I})$ is a reduction of $C(I)$.

Proof: By Theorem 2.4, $C(\hat{I})$ is a reduction of $C(I)$ if and only if for each $x \notin R(I)$, there exists a $k \in \hat{I}$ with $g_k(x) > 0$. But that is the same as saying that whenever $e(x) \neq \bar{0}$, there exists k with $y_k = 1$ and $e_k(x) = 1$; that is, $e(x)y^T \geq 1$. Thus, $C(\hat{I})$ is a reduction if and only if for all infeasible points x, $e(x)y^T \geq 1$; that is,

$$Ey \geq \bar{1}.$$

□

Next we give an example to illustrate the observations for a constraint set.

Example 2.1 Consider the set of one linear and two quadratic constraints given by

$$\begin{aligned} g_1(x) &= -x_1 + 2x_2 - 4 \\ g_2(x) &= x_1^2 + (x_2 - 1)^2 - 4 \\ g_3(x) &= x_1^2 + x_2^2 - 1 \end{aligned}$$

that are graphed in Figure 4.

In Figure 4, the three constraints partition $\mathbf{R}^2$ into 5 full dimensional regions. The points labeled x_1, x_2, …, x_5 are the observations. The binary word corresponding to x_1 is

$$e(x_1) = (e_1(x_1), e_2(x_1), e_3(x_1)) = (0, 1, 1),$$

since

$$g_1(x_1) < 0, \;\; g_2(x_1) > 0, \;\; g_3(x_1) > 0.$$

Similarly, we have

$$e(x_2) = (0, 0, 1), \; e(x_3) = (0, 0, 0), \; e(x_4) = (1, 0, 1), \; e(x_5) = (1, 1, 1).$$

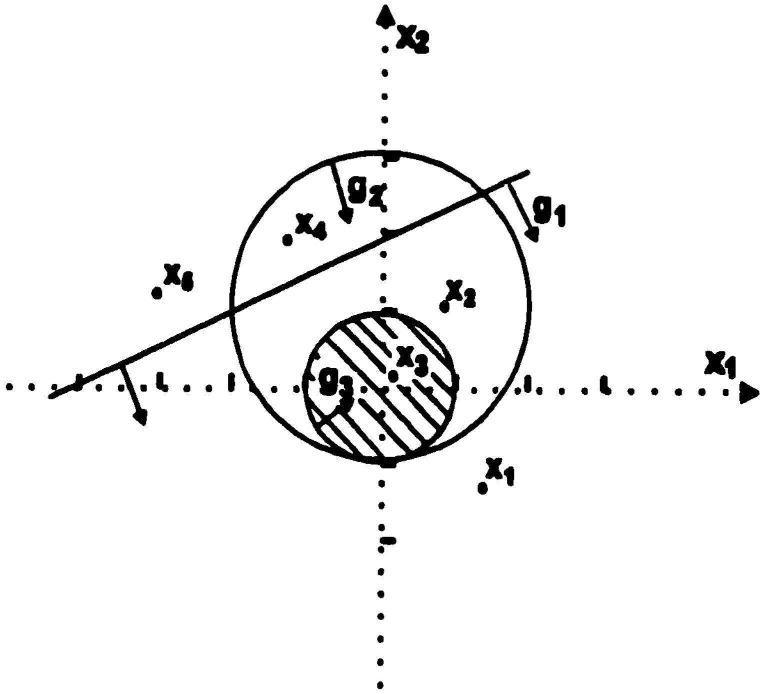

Figure 4: Observations for a set of one linear and two quadratic constraints

So, the corresponding SC matrix is

$$E = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The row $e(x_3)$ corresponding to the feasible point x_3 is not included in E.

□

2.5 The set covering matrix

We have seen the connection between the redundancy detection problem and the set covering feasibility problem $Ey \geq \bar{1}$. In this section, we will further explore the SC matrix E and reveal its connection to the problem of finding a Prime Representation(PR), as well as the problem of finding an Irreducible Infeasible System(IIS) and a Minimal Infeasible System(MIS).

We begin with some comments about the SC matrix E itself. First, we note that the number of rows in E is bounded by $2^m - 1$, the number of different, nonzero binary words of length m. However, for most systems of inequalities, the number of rows in E is much less [4].

In example 2.1, we had $m = 3$ so that the upper bound on the number of rows is $2^m - 1 = 7$. The SC matrix E had 4 rows. It is however, possible to have examples where the number of rows in E is $2^m - 1$.

Example 2.2 Consider the set of three linear constraints given by

$$\begin{aligned} g_1(x) &= x_1 + x_2 - 2 \\ g_2(x) &= -x_1 + x_2 - 2 \\ g_3(x) &= x_1^2 + (x_2 - 2)^2 - 4 \end{aligned}$$

that are graphed in Figure 5.

In Figure 5, the $m = 3$ constraints partition $\mathbf{R}^2$ into $2^m = 8$ full dimen-

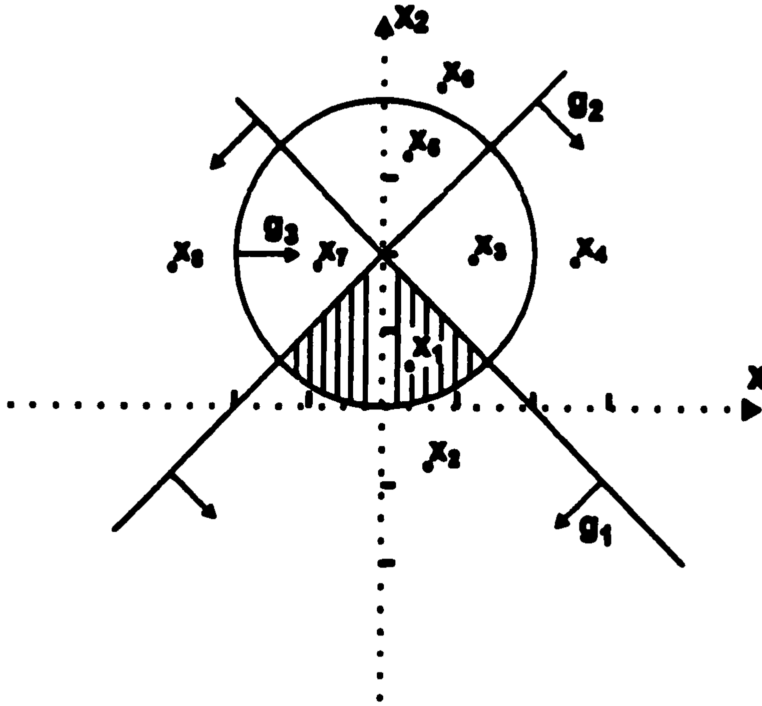

Figure 5: Observations for a set of three linear constraints

sional regions. As before we have

$$e(x_1) = (0,0,0), \quad e(x_2) = (0,0,1), \quad e(x_3) = (1,0,0), \quad e(x_4) = (1,0,1),$$
$$e(x_5) = (1,1,0), \quad e(x_6) = (1,1,1), \quad e(x_7) = (0,1,0), \quad e(x_8) = (0,1,1),$$

and the SC matrix is

$$E = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

□

It is easy to see that the sparsity of the matrix E having all $2^m - 1$ rows is about 50%. In fact, Boneh stated that one characterization of the matrix E corresponding to redundancy problems is that they have density close to 50% [4].

For the moment, let's consider E in the context of the SC feasibility problem. Let e_1 and e_2 be two distinct rows of E. If $e_1 \leq e_2$, then $e_1 y \geq 1$ implies that $e_2 y \geq 1$; that is, if one row of the matrix E dominates another row, that row is redundant and can be removed.

Definition 2.1 Let $\hat{E}$ be a submatrix of E such that (i) E and $\hat{E}$ both have m columns; and (ii) $\hat{E}$ has no redundant rows. We say that $\hat{E}$ is a *redundancy free reduction* of E.

We see that

$$\hat{E} = [\,0\;0\;1\,].$$

in Example 2.1. Interestingly, this gives $y = (0,0,1)^T$ as the only solution to $\hat{E}y \geq \bar{1}$. Thus, the only reduction of this system is obtained by removing constraints 1 and 2 as redundant, and keeping constraint 3 as necessary. Clearly, since $R(I) \neq \mathbf{R}^n$, this must be a minimal reduction, and $C(\{3\})$ is a minimal representation of $R(I)$.

We have seen that any solution to $Ey \geq \bar{1}$ gives a reduction $C(\hat{I})$ of $C(I)$. It makes sense to consider a reduction having the smallest number of constraints. This can be obtained by solving the set covering problem

$$\begin{array}{ll} \text{minimize} & c^T y \\ \text{s.t.} & Ey \geq \bar{1}. \\ & y \in \{0,1\}^m \end{array} \qquad (2)$$

If c is taken to be a vector of ones, the optimal solution will be a Prime Representation of $C(I)$. If $R(I)$ is empty, then an optimal solution to the SC problem gives an IIS. Other choices for c result in different solutions. For example, we may set c_i to be large for general nonlinear constraints, smaller for general linear constraints, and small for upper and lower bound constraints, so as to give preference to the simpler types of constraints.

Now, there is always some concern when modeling a problem as a set covering problem. This is because the general SC problem is known to be NP-hard [10]. However, let's look at the case of linear programming; that is, suppose that the g_i are linear. For each necessary constraint g_i, there exists a point x such that $e_j(x) = 1,\ j = i$, and $e_j(x) = 0,\ \forall j \in I \setminus \{i\}$. (We assume that $R(I)$ is full dimensional.) That is, we have the property that the redundancy free SC matrix $\hat{E}$ will only contain rows with a single "1". This, of course, results in an easily solved SC problem.

Example 2.3 Consider the system in Example 2.2, the SC matrix is

$$E = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

and the redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

So, the only, and, hence, optimal solution to the SC problem is $y = (1,1,1)^T$. That is, constraints $1,2$ and 3 are necessary, and $C(\{1,2,3\})$ is a Prime Representation of $R(I)$.

This property may also hold for other problems. For example, we had $\hat{E} = [001]$ for Example 2.1, so, $y = (0,0,1)^T$, that is, constraint 3 is necessary with constraints 1 and 2 redundant. This problem has nonlinear constraints.

While our experience shows that $\hat{E}$ will be sparse, it does not necessarily contain only rows having a single "1".

Example 2.4 Consider the set of four linear constraints given by

$$\begin{aligned} g_1(x) &= -x_1 - x_2 + 2 \\ g_2(x) &= -x_1 + x_2 + 1 \\ g_3(x) &= x_1 + x_2 - 1 \\ g_4(x) &= x_1 - x_2 + 1 \end{aligned}$$

that are graphed in figure 6, while this example has only linear constraints, we have $R(I) = \emptyset$ which is not full dimensional.

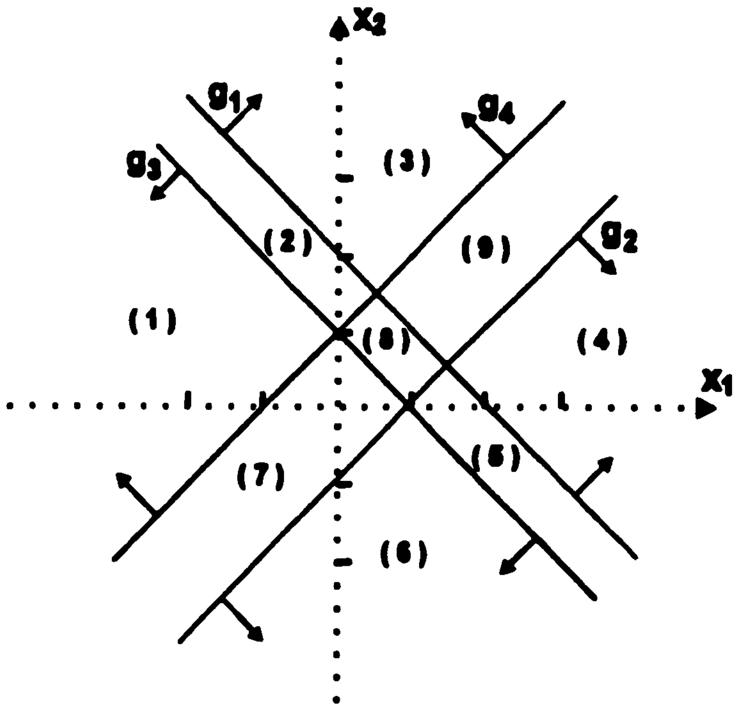

Figure 6: Observations for a set of four linear constraints

In Figure 6, the feasible region is empty. The SC matrix is

$$E = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

and the redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

□

So, we see that the SC model of the redundancy problem gives a unifying framework for MR and IIS problems, and no prior knowledge of feasibility is required. We now make a connection to the MIS problem.

Theorem 2.6 Assume that $R(I) = \emptyset$. Let e^* be a row of E having minimum cardinality in the sense that $e^*\bar{1} = \text{minimum } \{ e\bar{1} \mid e \text{ is a row of } E \}$. Set $\hat{I} = \{ i \mid e_i^* = 1 \}$. Then $C(\hat{I})$ is a MIS.

Proof: First we show $C(\hat{I})$ is an infeasibility set. Fix a point x with $e(x) = e^*$. Since $e_i^* = 1 \Leftrightarrow g_i(x) > 0$, $x \in R(I \setminus \hat{I})$. It follows from Definition 1.7 that $C(\hat{I})$ is an infeasibility set.

Second, since $e^{*}\bar{1} = \text{minimum } \{ e\bar{1} \mid e \text{ is a row of } E \}$, $C(\hat{I})$ has the minimum cardinality.

So, $\hat{I}$ indexes a minimal infeasibility set.

□

According to Chakravarti [8], the MIS problem is NP-hard. This is now easily understood, since in the worse case the number of rows in E is exponential in m.

So, we now see that the SC matrix provides a unifying framework for the PR, IIS and MIS problems. Unfortunately, it is not so easy to determine E, for a particular constraint set. The next section presents a probabilistic technique to determine some of the rows of E.

2.6 A probabilistic method to approximate E

We suggest the following probabilistic method for sampling points in $\mathbf{R}^n$, to generate rows of E.

The algorithm is based upon the assumption that for all rows e of the matrix E, there exists an x with $a\bar{1} \le x \le b\bar{1}$, $a, b \in \mathbf{R}, \bar{1} \in \mathbf{R}^n$, with $e(x) = e$.

	An algorithm to create E
Given:	An index set I and corresponding constraint set $C(I)$.
Initialization:	$j = 1$, $E \in \mathbf{R}^{0 \times m}$.
Repeat:	*Generate an observation.* Generate $z_k \sim U(0,1)$, and set $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$, where $x_{jk} = a + (b-a)z_k$. For all i, $i = 1, 2, \ldots, m$, evaluate $e(x_j)$ *Acceptance or Rejection:* If e_j is not an existing row of E, then replace E with $\left[\frac{E}{e}\right]$ (accept), otherwise, leave E unchanged (reject).
Termination:	If the stopping rule holds, output the SC matrix $E \in \mathbf{R}^{r \times m}$, else replace j with $j+1$ and repeat.

We first note that x_{jk} is uniformly distributed in the n dimensional hypercube $[a\bar{1}, b\bar{1}]$ since $x_{jk} = a + (b-a)z_k$ and $z_k \sim U(0,1)$. The stopping criteria may be the total number of iterations (e.g. 1000 iterations) or the number of iterations we went without adding a new row to the set covering matrix.

The formulation of the SC matrix E is the main implementation issue of the SC approach. El-Khatib [11] uses a modification of the HD algorithm to generate lines, $L(x_k, s_k)$, in $\mathbf{R}^n$. For each such line she collected the different binary words $e(x)$ generated by points one the line. She also removed redundant observations as part of the collection process. El-Khatib's approach was designed explicitly for linear constraints. Further Antic [1] includes new steps which (1) detects more redundancy in E during the HD process and (2) removes redundancies in matrix E after the HD process.

A look at the proof of theorem 2.3 shows that the hit-and-run algorithm finds points which violate only one constraint, i.e., they collect the observations such as $(0, 0, \ldots, 1, \ldots, 0)$. This connects the two approaches, SC and HD, for the case where $R(I) \neq \emptyset$.

2.7 Conclusion

We have shown that both the deterministic methods and the hit-and-run methods have limitations in the types of problem that can be handled. The advantage of Boneh's SC approach is that it can be used for very general constraint sets. In fact, Boneh stated that the only assumption required is that one can determine whether $g_i(x)$ is positive or nonpositive. A further advantage as is shown here, is that it gives a unifying framework for the PR, IIS and MIS problems.

The next chapter discusses an unexpected difficulty inherent in the probabilistic algorithms used to determine the matrix E.

3 An Unexpected Problem

3.1 Introduction

Since the creation of the matrix E depends on a random sampling of points in $\mathbf{R}^n$, we fully expect that the matrix E created by our algorithm will have missing rows. However, in Boneh's original paper, there is an implicit assumption that for each row of E, there is a nonzero probability of sampling a point in $\mathbf{R}^n$ that can generate that row. We use the example given by Boneh in his paper to show that this is not the case, that is that there is an unexpected problem. This was first noticed by McDonald [15]. This chapter contains the main contribution of the thesis, which is to explore this unexpected problem.

Example 3.1 ([4]) Consider the set of one linear and three quadratic constraints given by

$$
\begin{aligned}
g_1(x) &= -x_1 + x_2 - 3 \\
g_2(x) &= x_1(x_1 + x_2 - 1) \\
g_3(x) &= x_2(x_1 + x_2 - 1) \\
g_4(x) &= -x_1 x_2
\end{aligned}
$$

that are graphed in Figure 7.

These constraints define a partition of the $\mathbf{R}^2$ space into 11 full dimensional regions, including the feasible region, see Figure 7.

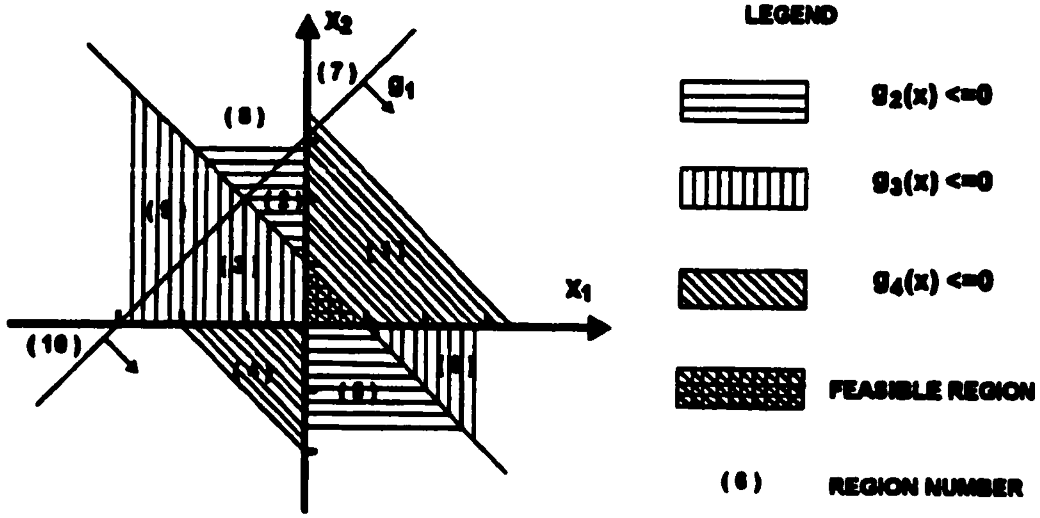

Figure 7: Set of one linear and three quadratic constraints

Based on the rows generated by observations in the regions, the SC matrix appears to be

$$E = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

The redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Consider the SC problem

$$\begin{array}{ll} \min & y_1 + y_2 + y_3 \\ \text{s.t.} & \hat{E}y \geq e,\ y \in \{0,1\}^4 \end{array}$$

There are three choices for an optimal solution to the SC problem. They are $y = (0,1,1,0)^T$, $y = (0,0,1,1)^T$ and $y = (0,1,0,1)^T$. The solution $y = (0,1,1,0)^T$ gives the reduction $C(\hat{I})$, $\hat{I} = \{2,\ 3\}$, with constraints 1 and 4 taken as redundant. Let's see what this means. Figure 8 is a graph of the constraints in $\hat{I}$.

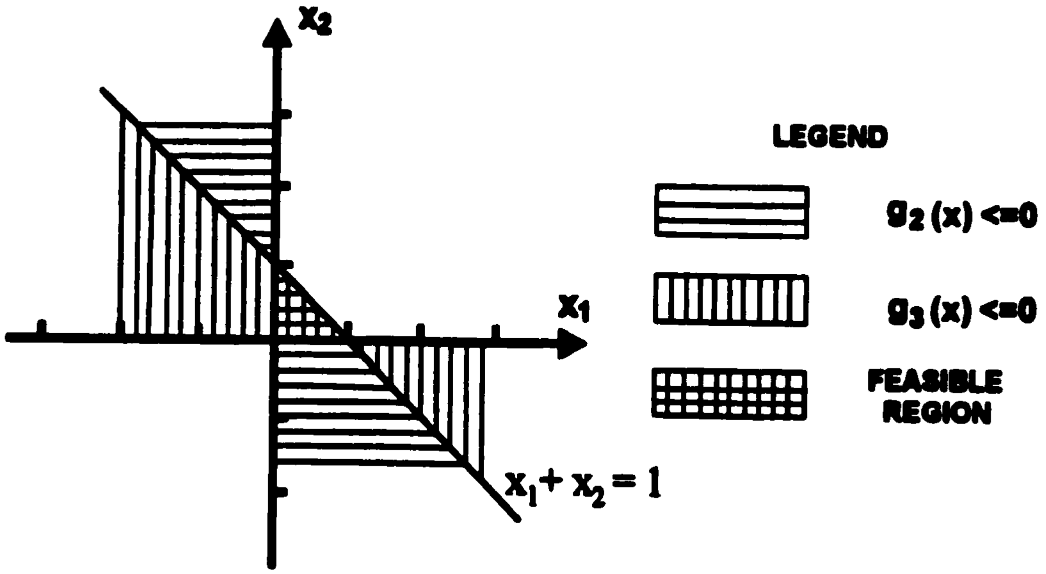

Figure 8: The incorrect reduction $C(\hat{I})$ of $C(I)$

Clearly, the feasible region now includes the entire line $x_1 + x_2 = 1$ that is not in the original feasible region, so that $C(\hat{I})$ is, in fact, not a reduction. The only source for this error is an incorrect SC matrix E.

In fact, there are rows of E that can only be generated by points on the constraint surfaces.

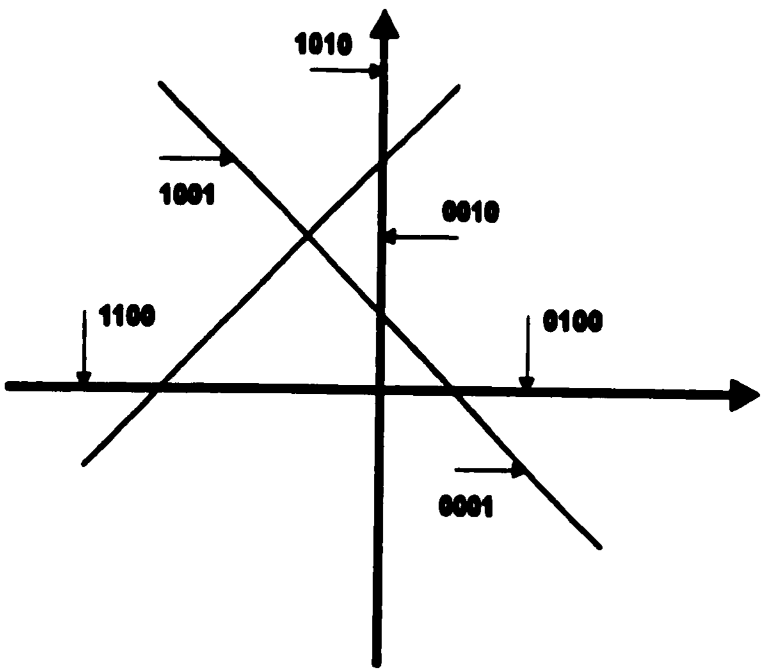

Figure 9: Rows of E on constraint surfaces

In Figure 9, we show the new rows obtained from the surface observations. So, the SC matrix is actually

$$E = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

with the redundancy free reduction

$$\hat{E} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The SC solution is $y = (0, 1, 1, 1)^T$ which corresponds with reduction $C(\hat{I})$, $\hat{I} = \{2, 3, 4\}$, with constraint 1 taken as redundant. This is correct.

□

In order to investigate this phenomenon, we need some new notation and definitions.

For any $x \in \mathbf{R}^n$, the index set I is partitioned as follows

$$\begin{aligned} I^+(x) &= \{i \in I | \; g_i(x) > 0\}, \\ I^-(x) &= \{i \in I | \; g_i(x) < 0\}, \\ I^0(x) &= \{i \in I | \; g_i(x) = 0\}. \end{aligned}$$

Definition 3.1 We say that $\hat{x}$ is a *regional point* if $|I^0(\hat{x})| = 0$, and a *surface point* if $|I^0(\hat{x})| \geq 1$.

Example 3.2 Consider the constraints

$$\begin{aligned} g_1(x) &= x_1 + x_2 - 2 \\ g_2(x) &= -x_1 + x_2 - 2 \\ g_3(x) &= -x_1 x_2 \end{aligned}$$

that are graphed in Figure 10.

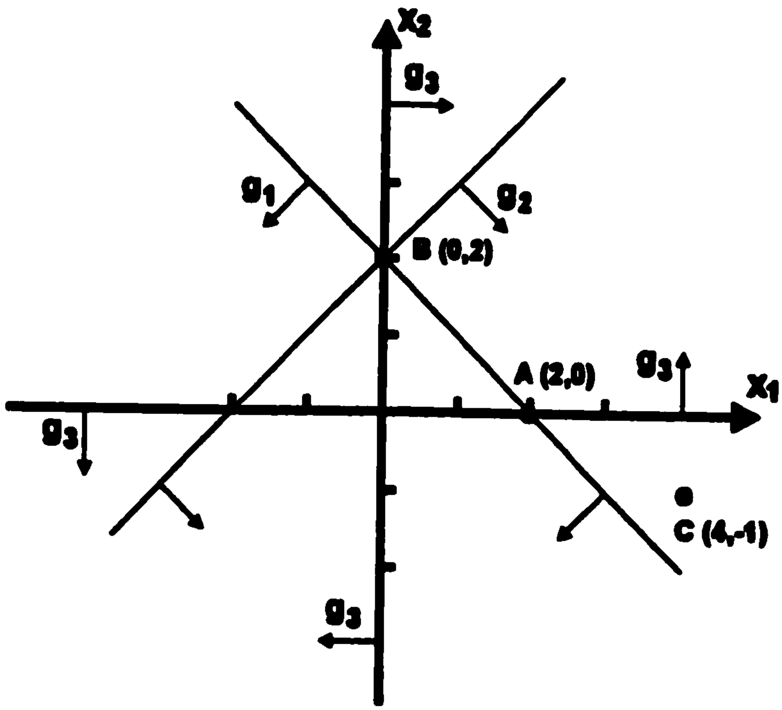

Figure 10: Regional and surface points

We have $A = (2, 0)$, $B = (0, 2)$, and $C = (4, -1)$. For the point A, we get the partition, $I^+(A) = \emptyset$, $I^-(A) = \{2\}$, $I^0(A) = \{1, 3\}$. For the point B, we get $I^+(B) = \emptyset$, $I^-(B) = \emptyset$, $I^0(B) = \{1, 2, 3\}$. For the point C, we get $I^+(C) = \{1, 3\}$, $I^-(C) = \{2\}$, $I^0(C) = \emptyset$. Since $|I^0(C)| = 0$, C is a regional point. Both A and B are surface points since $|I^0(A)| \geq 1$ and $|I^0(B)| \geq 1$.

It is interesting that while $B \in \{ x \mid g_i(x) = 0,\ i \in I^0(A) \}$, we have $I^0(B) \neq I^0(A)$.

□

Lemma 3.1 If $\hat{x}$ is a regional point, then there exists an $\epsilon > 0$, such that all $x \in B(\hat{x}, \epsilon)$ are regional points with $e(x) = e(\hat{x})$.

Proof: For each i, let

$$U_i = \begin{cases} \{ x \mid g_i(x) < 0 \} = R_i \setminus S_i & \text{if } g_i(\hat{x}) < 0 \\ \{ x \mid g_i(x) > 0 \} = R_i^c & \text{if } g_i(\hat{x}) > 0 \end{cases}$$

From continuity of g_i, U_i is open and $e_i(x) = e_i(\hat{x})$ for all $x \in U_i$. Thus

$$G = \bigcap_{i \in I} U_i$$

is also open and contains $\hat{x}$. Then $x \in G$ implies $e_i(x) = e_i(\hat{x})$ for all i, that is, $e(x) = e(\hat{x})$. So, $\exists \epsilon > 0$, $\mathbf{B}(\hat{x}, \epsilon) \subset G$ and $\forall x \in \mathbf{B}(\hat{x}, \epsilon)$, $e(x) = e(\hat{x})$.

□

If we take the cube $[a\bar{1}, b\bar{1}]$ large enough to intersect each of the regions G defined in the above way, then each region has non-zero probability of being chosen, since the points are chosen uniformly on that cube.

Lemma 3.2 The observed binary word corresponding with a regional point has nonzero probability of being sampled with the algorithm to collect observation randomly in $\mathbf{R}^n$.

Proof: We see this from Lemma 3.1.

□

Since it is quite likely that each surface $S_k, \forall k \in I$, has zero Lebesgue measure and hence zero probability of being sampled with the algorithm, we would like to know when the surface observations can be omitted without causing missing rows in E.

Definition 3.2 We say that there are *no surface observations* if for any $x \in S_k, i \in I$, there exists a regional point $\bar{x}$ with $e(\bar{x}) = e(\hat{x})$.

In the next section, we will give some examples in which we have surface observations.

3.2 When will a surface observation occur?

We present a few examples in which the surface observations contribute a row in the redundancy free reduction of the SC matrix. We state with the follow definition which will used by the examples.

Definition 3.3 We say that a point $\hat{x}$ is a *local minimizer* of a function g_i, if $\exists\epsilon > 0$, $\forall x \in \mathbf{B}(\hat{x}, \epsilon)$, $g_i(\hat{x}) \leq g_i(x)$. If also $|I^0(\hat{x})| \geq 1$, then $\hat{x}$ is a local minimizer surface point.

Example 3.3 Consider following a feasible system given by

$$\begin{array}{l} g_1(x) = ((x_1+1)^2 + (x_2+1)^2)(-x_1 - x_2 + 2) \\ g_2(x) = -x_1 - x_2 + 1 \end{array}$$

that are graphed in Figure 11, and in which the point $A(-1,-1)$ is a local minimizer surface point of the constraint function $g_1(x)$.

The two constraints define a partition of the $\mathbf{R}^2$ space into 3 full dimensional regions. The SC matrix appears to be

$$E = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

The redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

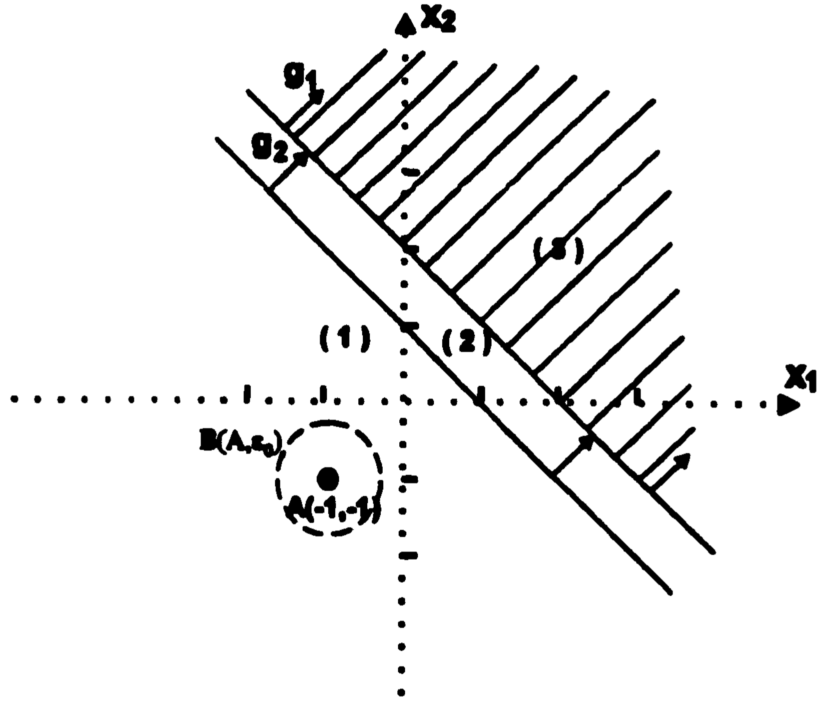

Figure 11: A feasible system with a local minimizer surface point

The optimal solution is $y = (1,0)^T$. It corresponds with the reduction $C(\hat{I})$, $\hat{I} = \{1\}$, with constraint 2 taken as redundant. Clearly, the feasible region only defined by constraint $g_1(x) \leq 0$ includes the point $A(-1,-1)$ which is not included in the original feasible region. The error is caused by an incorrect SC matrix because we missed the observation at the local minimizer surface point, $A(-1,-1)$ with $e(A) = (0,1)$.

The redundancy free reduction of the SC matrix is

$$E = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The optimal solution is $y = (1,1)^T$ which means that constraint 1 and 2 are necessary.

□

Example 3.4 Consider the following infeasible system given by

$$\begin{aligned} g_1(x) &= ((x_1+1)^2 + (x_2+1)^2)(-x_1 - x_2 + 2) \\ g_2(x) &= -x_1 + x_2 + 1 \\ g_3(x) &= x_1 + x_2 - 1 \end{aligned}$$

that are graphed in Figure 12, and in which the point $A(-1,-1)$ is a local minimizer surface point of the constraint function $g_1(x)$.

The three constraints define a partition of the $\mathbf{R}^2$ space into 6 full dimensional regions. The SC matrix appears to be

$$E = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

The redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

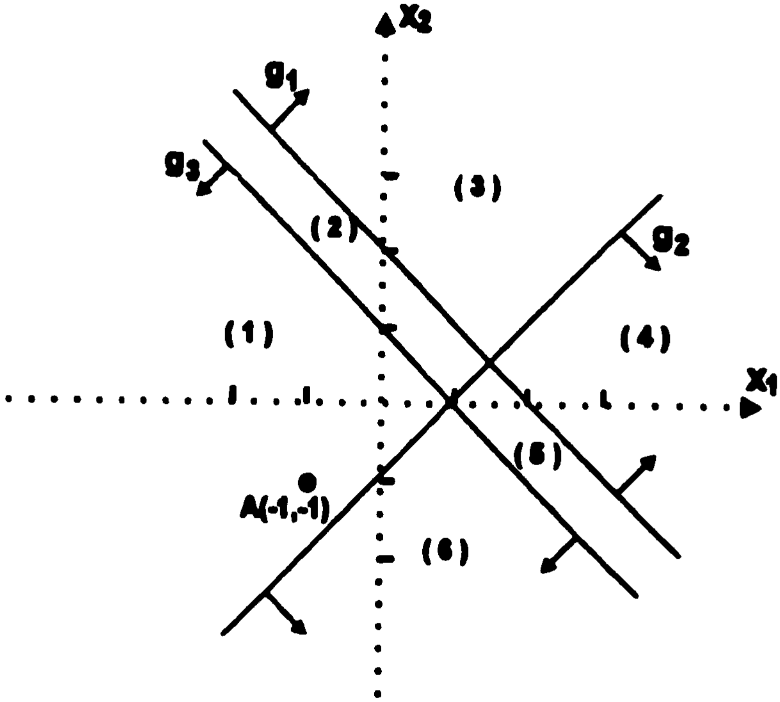

Figure 12: An infeasible system with a local minimizer surface point

The optimal solution is $y = (1, 0, 1)^T$. It corresponds with the reduction $C(\hat{I})$, $\hat{I} = \{1, 3\}$, with constraint 2 taken as redundant. Clearly, the system becomes feasible if constraint 2 is removed, the point $A(-1, -1)$ becomes a feasible point. The error is caused by an incorrect SC matrix because we missed the observation at the local minimizer surface point $A(-1, -1)$ with $e(A) = (0, 1, 0)$.

The redundancy free reduction of the SC matrix is

$$E = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The optimal solution is $y = (1, 1, 1)^T$ which classifies constraints 1, 2 and 3 as necessary.

□

Next we will give a definition that is a general form of a local minimizer surface point and that plays a key role in exploring the unexpected problem.

Definition 3.4 We say that the surface point $\hat{x}$ is a *local quasi-minimizer* if

$$\hat{x} \notin \text{cl}(\{x \mid g_i(x) < 0,\ i \in I^0(\hat{x})\}.$$

Actually,

$$\hat{x} \notin \text{cl}(\{x \mid g_i(x) < 0,\ i \in I^0(\hat{x})\}$$

$$\Longleftrightarrow \exists \epsilon > 0,\ \forall x \in \mathbf{B}(\hat{x}, \epsilon),\ \exists i \in I^0(\hat{x}),\ \text{such that } g_i(x) \geq 0.$$

$$\Longleftrightarrow \exists \epsilon > 0,\ \forall x \in \mathbf{B}(\hat{x}, \epsilon),\ \max_{i \in I^0(\hat{x})} g_i(x) \geq 0.$$

Since $g_i(\hat{x}) = 0$, for all $i \in I^0(\hat{x})$, the point $\hat{x}$ is a local minimizer of the function

$$\max_{i \in I^0(\hat{x})} g_i(x).$$

We can see that a local minimizer surface point is also a local quasi-minimizer.

Example 3.5 Consider the following feasible system given by

$$\begin{aligned} g_1(x) &= -x_1^2 + x_2^2 + 4 \\ g_2(x) &= (x_1+1)^2 + x_2^2 - 9 \\ g_3(x) &= x_1 - x_2 \end{aligned}$$

that are graphed in Figure 13, and in which the point $A = (2,0)$ is a local quasi-minimizer.

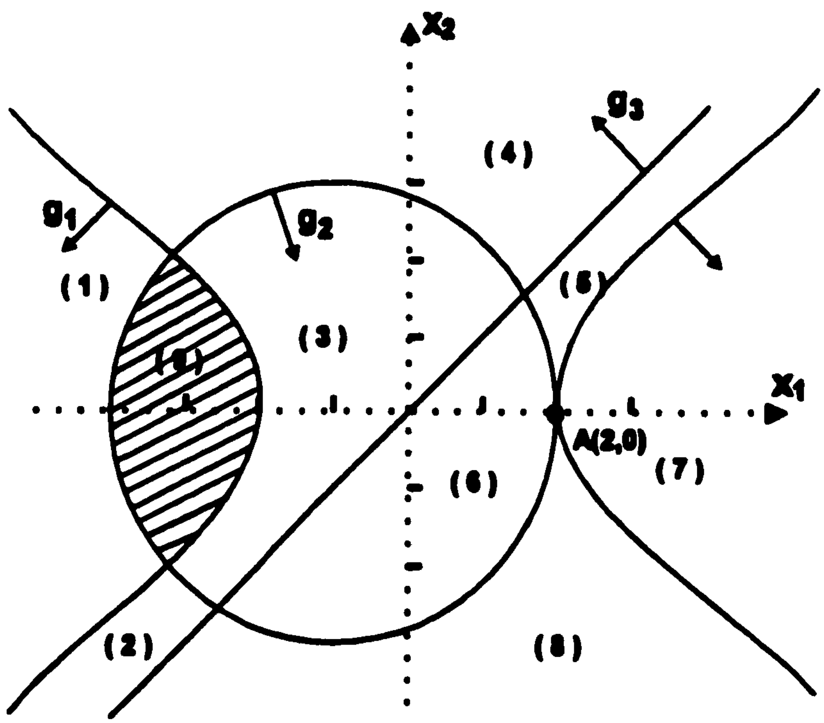

Figure 13: A feasible system with a local quasi-minimizer surface point

The three constraints define a partition of the $\mathbf{R}^2$ space into 9 full dimensional regions. The SC matrix appears to be

$$E = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

The redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Consider the optimal solution $y = (1, 1, 0)^T$. It corresponds with a reduction $C(\hat{I})$, $\hat{I} = \{1, 2\}$, with constraint 3 taken as redundant. Clearly, the feasible region defined by constraint 1 and 2 includes the point $A(2, 0)$ which is not in the original feasible region. The error is caused by an incorrect SC matrix because we missed the observation at the local quasi-minimizer A with $e(A) = (0, 0, 1)$.

The SC matrix is actually

$$E = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The optimal solution is $y = (1, 1, 1)^T$ which classifies constraints 1, 2 and 3 as necessary.

□

Example 3.6 Consider the following infeasible system given by

$$\begin{aligned} g_1(x) &= -x_1^2 + x_2^2 + 4 \\ g_2(x) &= (x_1 - 0.5)^2 + y^2 - 2.25 \\ g_3(x) &= x_1 - x_2 \end{aligned}$$

that are graphed in Figure 14, and in which the point $A = (2,0)$ is a local quasi-minimizer.

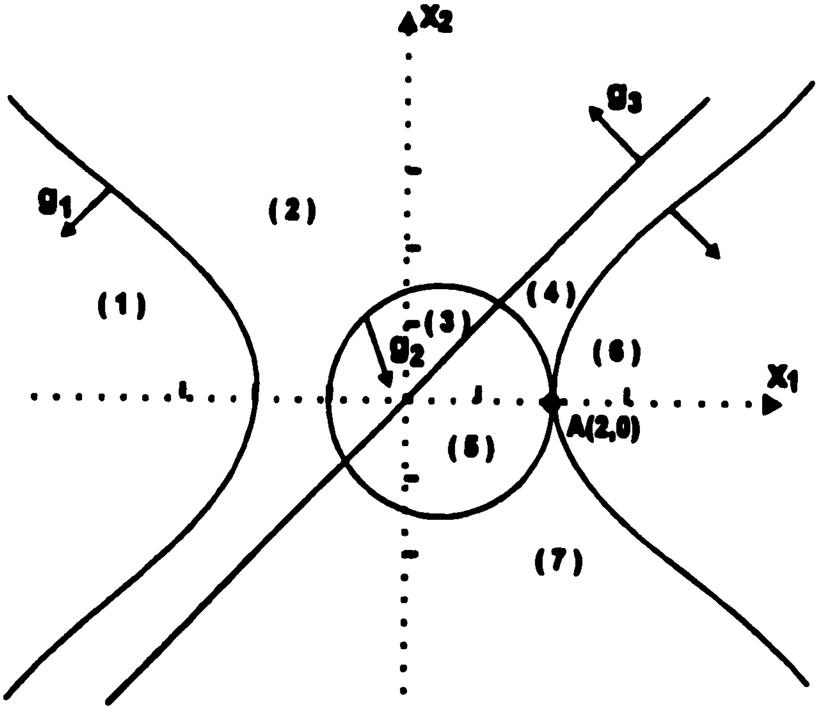

Figure 14: An infeasible system with a local quasi-minimizer

The three constraints define a partition of the $\mathbf{R}^2$ space into 7 full dimensional regions. The SC matrix appears to be

$$E = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

The redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Consider the optimal solution $y = (1,1,0)^T$. It corresponds with reduction $C(\hat{I})$, $\hat{I} = \{1,2\}$, with constraint 3 taken as redundant. Clearly, the system becomes feasible since $A(2,0)$ satisfies both constraints 1 and 2. The error is caused by an incorrect SC matrix because we missed the observation at the local quasi-minimizer $A(2,0)$ with $e(A) = (0,1,0)$.

The redundancy free SC matrix is

$$\hat{E} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The optimal solution is $y = (1,1,1)^T$ which classifies constraint 1,2 and 3 as necessary.

□

. Lastly, let's recall the original Boneh example (refer to Figure 7). The incorrect SC matrix is caused by our missing the observation at the local quasi-minimizer (refer to Figure 9). We can see the surface point corresponding with the observation binary word $(0,1,0,0)$, $(0,0,0,1)$, $(1,1,0,0)$, $(1,0,0,1)$, $(1,0,1,0)$ and $(0,0,1,0)$ are all local quasi-minimizers.

We have shown that missing observations at a local quasi-minimizer can cause an incorrect SC matrix. In section 3.4, we will further conclude that only local quasi-minimizers cause the problem. We will first show the relationship between the SC matrix and observations at regional and surface points in next section.

3.3 Where is the information?

Theorem 3.1 Let $\hat{x} \in \mathbf{R}^n$ and let C be the intersection of all surfaces containing $\hat{x}$, i.e.,

$$C = \bigcap_{i \in I^0(\hat{x})} S_i.$$

If $I^0(\hat{x}) = \emptyset$, then $C = \mathbf{R}^n$. If there exists S_j, $j \notin I^0(\hat{x})$ that intersects C then there exists an $\bar{x} \in S_l \cap C$, $l \notin I^0(\hat{x})$ such that $e(\hat{x}) \geq e(\bar{x})$; that is, $e(\hat{x})$ is redundant and is not needed as a row in the redundancy free reduction of the SC matrix.

Proof: Since the number of constraint functions is finite and since they are continuous, the set

$$S = \bigcup_{j \notin I^0(\hat{x})} S_j$$

is closed. Let $\bar{x}$ be such that $||\hat{x} - \bar{x}|| = \text{minimum } \{||\hat{x} - x|| : x \in (S \cap C)\}$. For all $x \in C \cap \mathbf{B}(\hat{x}, ||\hat{x} - \bar{x}||)$, we have that $e(x) = e(\hat{x})$, i.e., if $g_i(\hat{x}) \leq 0$ then $g_i(x) \leq 0$. It then follows from continuity and the definition of $\bar{x}$ that if $g_i(\hat{x}) \leq 0$ then $g_i(\bar{x}) \leq 0$ for all $i \in I$. Thus, $e(\hat{x}) \geq e(\bar{x})$.

□

Since $\bar{x} \in S_k, k \in I$, we can see that all the information about the SC matrix can be obtained from surface points.

3.4 When and where the set covering approach works

In this section, we determine conditions under which the surface observation are not needed in order to obtain the redundancy free reduction of the SC matrix.

Theorem 3.2 Let $\hat{x} \in S_k$ be such that $|I^0(\hat{x})| \geq 1$. If $\hat{x}$ is not a local quasi-minimizer, then there exist an $\bar{x}$ with $|I^0(\bar{x})| = 0$, such that $e(\bar{x}) = e(\hat{x})$.

Proof: Let $\hat{x} \in S_k$.

Suppose we removed the constraint $g_i(x)$, $i \in I^0(\hat{x})$, then $\hat{x}$ becomes a regional point, and there exists an $\epsilon_1 > 0$, such that for all

$$x \in \mathbf{B}(\hat{x}, \epsilon_1)$$

are regional points and $e(x) = e(\hat{x})$ in $R(I \setminus I^0(\hat{x}))$.

On the other hand, from the Definition 3.4, if $\hat{x}$ is not a local quasi-minimizer, then

$$\hat{x} \in \mathrm{cl}(\{x \mid g_i(x) < 0, \ \forall i \in I^0(\hat{x})\}),$$

i.e.,

$$\forall \epsilon_2 > 0, \ \ \mathbf{B}(\hat{x}, \epsilon_2) \cap \{x \mid g_i(x) < 0, \ \forall i \in I^0(\hat{x})\} \neq \emptyset.$$

W.L.O.G., let $\epsilon_2 = \epsilon_1/2$ and

$$\bar{x} = \mathbf{B}(\hat{x}, \epsilon_2) \cap \{x \mid g_i(x) < 0, \ \forall i \in I^0(\hat{x})\},$$

It is clear that $\bar{x} \in \mathbf{B}(\hat{x}, \epsilon_1)$, so $\bar{x}$ is a regional point and $e(\bar{x}) = e(\hat{x})$ in $R(I \setminus I^0(\hat{x}))$, i.e.,

$$e_j(\bar{x}) = e_j(\hat{x}), \ \ \forall j \in I \setminus I^0(\hat{x}).$$

Since $\bar{x}$ is not in any surface defined by $g_i(x)$, $i \in I^0(\hat{x})$, so $\bar{x}$ is a regional point in $R(I)$ and $I^0(\bar{x}) = 0$. Since

$$e_j(\bar{x}) = 0, \ \ \forall j \in I^0(\hat{x})$$

because $\bar{x} \in \{ x \mid g_i(x) < 0, \forall i \in I^0(\hat{x}) \}$, and

$$e_j(\hat{x}) = 0, \quad \forall j \in I^0(\hat{x}),$$

because $g_j(\hat{x}) = 0, \forall j \in I^0(\hat{x})$, so,

$$e(\bar{x}) = e(\hat{x}).$$

□

This theorem states that we needn't worry about the surface observations if they are not local quasi-minimizers. So, we can get the correct SC matrix with probability one using random sampling algorithm.

We note that it is not necessary that the local quasi-minimizer contribute a row in the redundancy free SC matrix.

Example 3.7 Consider the following feasible system given by

$$g_1(x) = -x_1 - x_2 + 2$$
$$g_2(x) = ((x_1 + 1)^2 + (x_2 + 1)^2)(-x_1 - x_2 + 1)$$

that are graphed in Figure 15, and in which the point $A(-1, -1)$ is a local quasi-minimizer.

It is clear that $(-1, -1)$ is a local quasi-minimizer since $I^0(A) = \{2\}$ and $A \notin \{ x \mid g_2(x) < 0 \}$. The two constraints define a partition of the $\mathbf{R}^2$ space

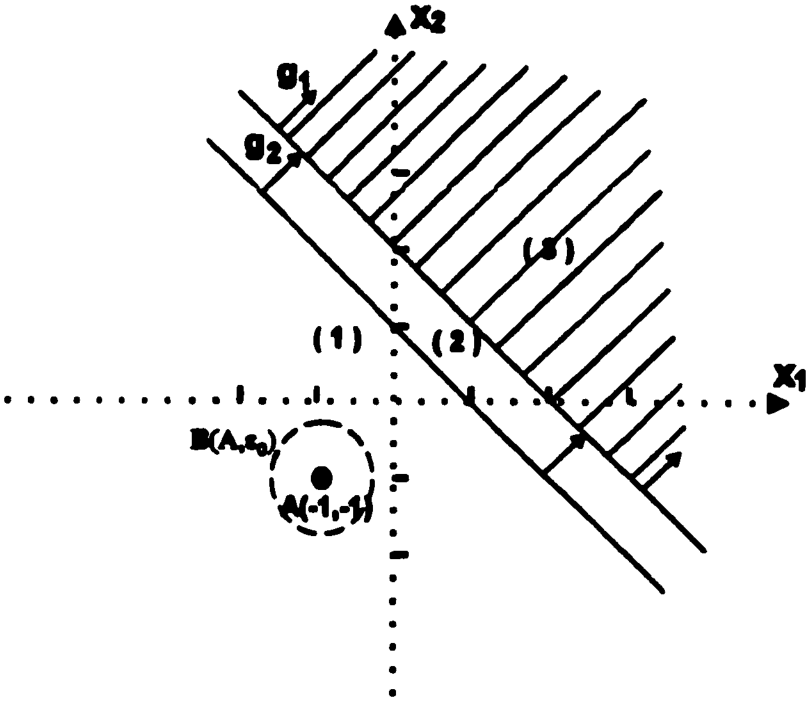

Figure 15: A local quasi-minimizer without the new observation

into 3 regions. The local quasi-minimizer $(-1,-1)$ has the observation $(1,0)$ that is same as the observation at any point in the full dimensional region (2). We have SC matrix

$$E = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

The redundancy free reduction is

$$\hat{E} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

3.5 Conclusion

In this chapter, we gave a few examples of feasible and infeasible systems to show that the SC matrix obtained without the observations at the local quasi-minimizer point may be incorrect. Also we presented the relationship between the SC matrix and observations at surface and regional points. Lastly, we derived the condition under which the surface observations are not needed to contribute rows to the redundancy free reduction of the SC matrix using a random sampling algorithm. In the next Chapter, we will apply the result of this Chapter to convex systems.

4 Convex Constraints

In this chapter, we assume that the functions $g_i, i \in I$ are convex, and that there are no implicit equalities in the constraint system.

We start with a review of basic results on convexity.

Lemma 4.1 ([17]) For any convex set $C \subseteq \mathbf{R}^n$, $\text{cl}(C)$ is convex.

Lemma 4.2 ([17]) Let $C \subseteq \mathbf{R}^n$ be a convex set, $x \in \text{int}(C)$ and $y \in \text{cl}(C)$, then $(1-\lambda)x + \lambda y \ \in \text{int}(C)$ for $0 \leq \lambda < 1$, i.e., $[x, y) \subset C$.

Lemma 4.3 ([17]) If $g_i(x)$ is a convex function in $\mathbf{R}^n$, then R_i and $R_i \setminus S_i$ are convex sets.

Proof: We give a proof for R_i. The proof for $R_i \setminus S_i$ is similar. If R_i contains one or no points, the Lemma is true.

If x_1 and x_2 are two points in R_i, then

$$g_i(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g_i(x_1) + (1-\lambda)g_i(x_2) \leq \lambda 0 + (1-\lambda)0 = 0.$$

Thus, $\lambda x_1 + (1-\lambda)x_2 \in R_i$, R_i is a convex set.

□

First we present a Lemma for a single constraint that is required by the Theorem later.

Lemma 4.4 Consider constraint i. The set $(R_i \setminus S_i) \neq \emptyset$ and $R_i = \text{cl}(R_i \setminus S_i)$.

Proof: First we show that $(R_i \setminus S_i) \neq \emptyset$. Suppose $(R_i \setminus S_i) = \emptyset$ then $R_i = S_i$, this means that the constraint i is an implicit inequality constraint. So $(R_i \setminus S_i) \neq \emptyset$.

Now we show that $R_i = \text{cl}(R_i \setminus S_i)$. First we show $R_i \subseteq \text{cl}(R_i \setminus S_i)$. It is obvious that $\forall x \in (R_i \setminus S_i)$, $x \in \text{cl}(R_i \setminus S_i)$.

Consider $\hat{x} \in S_i$. Suppose $\hat{x} \notin \text{cl}(R_i \setminus S_i)$, i.e., $\exists \epsilon > 0$, $\forall x \in \mathbf{B}(\hat{x}, \epsilon)$, such that $g_i(x) \geq 0$.

Let $\bar{x} \in (R_i \setminus S_i)$, i.e., $g_i(\bar{x}) < 0$. Let $l = ||\hat{x} - \bar{x}||$, $\lambda = 1 - \frac{\epsilon}{2l}$, $\check{x} = \lambda\hat{x} + (1-\lambda)\bar{x}$, then

$$||\check{x} - \hat{x}|| = ||\lambda\hat{x} + (1-\lambda)\bar{x} - \hat{x}|| = (1-\lambda)||\hat{x} - \bar{x}|| = \frac{\epsilon}{2l}l = \frac{\epsilon}{2}.$$

So, $\check{x} \in \mathbf{B}(\hat{x}, \epsilon)$, and

$$g_i(\check{x}) = g_i(\lambda\hat{x} + (1-\lambda)\bar{x}) \leq \lambda g_i(\hat{x}) + (1-\lambda)g_i(\bar{x}) < 0. \tag{3}$$

This contradicts that $\forall x \in \mathbf{B}(\hat{x}, \epsilon)$, $g_i(x) \geq 0$.

So, $\hat{x} \in \text{cl}(R_i \setminus S_i)$. Since $\hat{x}$ is arbitrary, $R_i \subseteq \text{cl}(R_i \setminus S_i)$.

Second, we show $\text{cl}(R_i \setminus S_i) \subseteq R_i$. We only need show that there isn't $\hat{x} \in \text{cl}(R_i \setminus S_i)$ such that $g_i(\hat{x}) > 0$.

Suppose $g_i(\hat{x}) > 0$, then $\exists \epsilon > 0$, $\forall x \in \mathbf{B}(\hat{x}, \epsilon)$ such that $g_i(x) > 0$. This means that $\hat{x} \notin \mathrm{cl}(R_i \setminus S_i)$. So, $\mathrm{cl}(R_i \setminus S_i) \subseteq R_i$.

So, $R_i = \mathrm{cl}(R_i \setminus S_i)$.

□

Lemma 4.5 Consider $\hat{x}$, $I^0(\hat{x}) = \{i_1, i_2, \ldots, i_t\}$, let

$$\mathcal{R}_t = \{\, x \mid g_{i_j}(x) < 0,\ \forall i_j \in I^0(\hat{x})\}$$

then we have $\mathcal{R}_t \neq \emptyset$ and $R(I^0(\hat{x})) = \mathrm{cl}(\mathcal{R}_t)$.

Proof: First, we define

$$\mathcal{R}_l = \{\, x \mid g_{i_j}(x) < 0,\ j = 1, \ldots, l.\}$$

We use the method of induction. Consider $l = 1, 2, \ldots, t$.

1) When $t = 1$, from Lemma 4.4, $\mathcal{R}_1 \neq \emptyset$ and $R_{i_1} = \mathrm{cl}(\mathcal{R}_1)$.

2) When $l < t$, suppose that

$$\mathcal{R}_l \neq \emptyset \text{ and } R(I^0(\hat{x})) = \mathrm{cl}(\mathcal{R}_l)$$

3) First, we only consider the first $t-1$ constraints.

From 2),we know that

$$\mathcal{R}_{t-1} \neq \emptyset \text{ and } R(I^0(\hat{x}) \setminus \{i_t\}) = \mathrm{cl}(\mathcal{R}_{t-1}).$$

It is clear that $(R_{i_t} \setminus S_{i_t}) \neq \emptyset$, otherwise, constraint i_t is an implicit equality.

Suppose $\mathcal{R}_t = \emptyset$. First we show that $S_{i_t} \cap \mathcal{R}_{t-1} = \emptyset$.

Suppose that there exists $\bar{x} \in S_{i_t} \cap \mathcal{R}_{t-1}$. We can find $\epsilon_1 > 0$, such that for all $x \in \mathbf{B}(\bar{x}, \epsilon_1)$, $x \in \mathcal{R}_{t-1}$. From Lemma 4.4, $\bar{x} \in \mathrm{cl}(R_{i_t} \setminus S_{i_t})$. For all $\epsilon_2 > 0$ and $\epsilon_2 \leq \epsilon_1$, such that there exists $y \in \mathbf{B}(\bar{x}, \epsilon_2)$ and $g_{i_t}(y) < 0$.

Since $y \in \mathbf{B}(\bar{x}, \epsilon_2) \subseteq \mathbf{B}(\bar{x}, \epsilon_1)$, so

$$y \in (R_{i_t} \setminus S_{i_t}) \cap \mathcal{R}_{t-1} = \mathcal{R}_t.$$

This contradicts that $\mathcal{R}_t = \emptyset$.

Second we show that $(R_{i_t} \setminus S_{i_t}) \cap (\mathrm{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1}) = \emptyset$.

Suppose that there exists $\bar{x} \in (R_{i_t} \setminus S_{i_t}) \cap (\mathrm{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1})$.

We can find $\epsilon_1 > 0$, such that for all $x \in \mathbf{B}(\bar{x}, \epsilon_1)$, $g_{i_t}(x) < 0$. Also, since $\bar{x} \in \mathrm{cl}(\mathcal{R}_{t-1})$, we can find $\epsilon_2 > 0$ and $\epsilon_2 \leq \epsilon_1$, such that there exists $y \in \mathbf{B}(\bar{x}, \epsilon_2)$ and $y \in \mathcal{R}_{t-1}$.

Since $y \in \mathbf{B}(\bar{x}, \epsilon_2) \subseteq \mathbf{B}(\bar{x}, \epsilon_1)$, so

$$y \in (R_{i_t} \setminus S_{i_t}) \cap \mathcal{R}_{t-1}.$$

This contradicts that $\mathcal{R}_t = \emptyset$. So,

$$(R_{i_t} \setminus S_{i_t}) \cap (\mathrm{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1}) = \emptyset.$$

So,

$$R_{i_t} \cap (\text{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1})$$

$$= (S_{i_t} \cap (\text{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1})) \cup ((R_{i_t} \setminus S_{i_t}) \cap (\text{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1}))$$

$$= S_{i_t} \cap (\text{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1}).$$

It is obvious that

$$R_{i_t} \cap \mathcal{R}_{t-1} = (S_{i_t} \cap R_{t-1}) \cup ((R_{i_t} \setminus S_{i_t}) \cap \mathcal{R}_{t-1})$$

So,

$$R(I^0(\hat{x})) = (R_{i_t} \cap R(I^0(\hat{x}) \setminus \{i_t\})) = (R_{i_t} \cap \text{cl}(\mathcal{R}_{t-1}))$$

$$= (R_{i_t} \cap (\text{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1})) \cup (R_{i_t} \cap \mathcal{R}_{t-1})$$

$$= S_{i_t} \cap (\text{cl}(\mathcal{R}_{t-1}) \setminus \mathcal{R}_{t-1})$$

This means that $g_{i_t}(x) \le 0$ is an implicit equality.

So, we proved that $\mathcal{R}_t \neq \emptyset$.

Let $\bar{x} \in \mathcal{R}_t$. Next we will show $R(I^0(\hat{x})) = \text{cl}(\mathcal{R}_t)$.

First we show that $R(I^0(\hat{x})) \subseteq \text{cl}(\mathcal{R}_t)$.

By (3), if $x \in \mathcal{R}(I^0(\hat{x}))$, then

$$[\bar{x}, x) \subset \bigcap_{i \in I^0(\hat{x})} (R_i \setminus S_i) = \mathcal{R}_t.$$

Therefore $x \in \text{cl}(\mathcal{R}_t)$.

Second, we show $\text{cl}(\mathcal{R}_t) \subseteq R(I^0(\hat{x}))$.

We only need show that there isn't $\hat{x} \in \text{cl}(\mathcal{R}_t)$ such that $g_i(\hat{x}) > 0$, $i \in I^0(\hat{x})$.

Suppose $g_i(\hat{x}) > 0$, then $\exists \epsilon > 0$, $\forall x \in \mathbf{B}(\hat{x}, \epsilon)$ such that $g_i(x) > 0$. This means that $\hat{x} \notin \text{cl}(R_i \setminus S_i)$, i.e., $\hat{x} \notin \text{cl}(\mathcal{R}_t)$. So, $\text{cl}(\mathcal{R}_t) \subseteq R(I^0(\hat{x}))$.

So, $\text{cl}(R_t) = R(I^0(\hat{x}))$.

□

Theorem 4.1 If the g_i, for all $i \in I$, are convex, then there are no local quasi-minimizers in this system.

Proof: Since for each surface point $\hat{x}$ with $|I^0(\hat{x})| = t$, $\hat{x} \in R(I^0(\hat{x}))$, from Lemma 4.5, $R(I^0(\hat{x})) = \text{cl}(\mathcal{R}_t)$, so, $\hat{x} \in \text{cl}(\mathcal{R}_t)$. By Definition 3.4, $\hat{x}$ is not a local minimizer.

□

We conclude from Theorem 3.2 and 4.1 that we can find a regional point $\bar{x}$, such that $e(\bar{x}) = e(\hat{x})$. That is, we needn't worry about the surface observations and can get the correct SC matrix with probability one using random sampling algorithm.

5 Conclusion

In this thesis we presented Boneh's Set Covering (SC) approach to redundancy detection. As one contribution, we showed that the SC approach can serve as a unifying framework for the Prime Representation (PR), the Irreducible Infeasible Set (IIS) problem, and the Minimal Infeasibility Set (MIS) problem. Typically, these have been thought of a three distinct, although related, problems.

The major contribution of this thesis has to do with rows of the SC matrix E that have probability zero of being detected by our probabilistic algorithm. We have shown that such rows correspond to points that must be the local quasi-minimizers, a concept introduced in the thesis. We also provided an example to show that even though a point is a local quasi-minimizer, the corresponding row in E may still have positive detection probability. We also proved that all rows of E can be determined by surface observations alone. Finally we showed that systems of convex constraints, which includes linear systems, have no local quasi-minimizers.

At last, we remark that in this thesis we assume that the constraint functions are continuous, there are not any implicit equality constraints.

As future research direction we recommend:

1. The characterizations of other systems without local quasi-minimizers.

2. The development of methods that can collect surface observations.

References

[1] M. Antic, "Redundancy and infeasibility with a set covering model", Master Thesis, University of Windsor(Windsor, ON, 1998).

[2] H. Berbee, C.C. Boender, A.R. Kan, C. Sheffer, R. Smith and T. Telgen, "Hit-and-run algorithms for the identification of nonredundant linear inequalities", Mathematical Programming 37(1987)184-207.

[3] A. Boneh, "PREDUCE - A probabilistic algorithm identifying redundancy by a random feasible point generator (RFPG)", In: Karwan et al. [13]pp. 108-132.

[4] A. Boneh, "Identification of redundancy by a set-Covering equivalence", Operational Research'84 407-422.

[5] A. Boneh, R.J. Caron, F.W. Lemire, J.F. Mcdonald, J. Telgen and T. Vorst, "Note on prime representations of convex polyhedral sets", Journal of Optimization Theory and Applications 1(1989)137-142.

[6] J.C.G. Boot, "On trivial and binding constraints in programming problems", Management Science 8(1962)419-441.

[7] R.J. Caron, J.F. McDonald and C.M. Ponic, "A degenerate extreme point strategy for the classification of linear constraints as redundant or necessary", Journal of Optimization Theory and Applications 2(1989)225-237.

[8] N. Chakravarti, "Some results concerning post-infeasibility analysis", European Journal of Operational Research 73(1994)139-143.

[9] J.W. Chinneck and E.W. Dravnieks, "Locating minimal infeasible constraint sets in linear programs", ORSA Journal on Computing 3(1991)157-167.

[10] V. Chvatal, "A greedy heuristic for the set covering problem", Mathematics of Operations Research 4(1979)233-235.

[11] H. M. El-Khatib, "A probabilistic method for cleaning contaminated systems of linear inequalities", Master Thesis, University of Windsor(Windsor, ON, 1997).

[12] A.V. Fiacco and G.P. McCormick, Nonlinear programming: Sequential Unconstrained Minimization Techniques (Society for Industrial and Applied Mathematics, 1990).

[13] M.H. Karwan, V. Lotfi, J. Telgen and S. Zionts, Redundancy in mathematical programming (Springer-Verlag, Berlin, 1983).

[14] D.E. Knuth, The art of computer programming, (Addison-wesley publishing company, 1969).

[15] J.F. McDonald, private communication with A. Boneh (1987).

[16] W.T. Obuchowska and R.J. Caron, "Minimal representation of quadratically constrained convex feasible regions", Mathematical Programming 68(1995)169-186.

[17] R.T. Rockafellar, Convex Analysis, (Princeton University Press, Princeton, NJ, 1970).

[18] J. Telgen, "Minimal representation of convex polyhedral sets", Journal of Optimization Theory and Applications 38(1982)1-24.

Vita Auctoris

NAME :	Jinghua Feng
PLACE OF BIRTH	Linchuan, P.R. China
YEAR OF BIRTH	1972
EDUCATION	Linchuan Second High School, Linchuan, P.R. China 1983 – 1989
	Xidian University, Xi'an, P.R. China 1989 – 1993 B.Sc. Applied Mathematics(major), Computer Science (minor)
	Nanchang University, Nanchang, P.R. China 1995 – 1998 M.A.Sc. Power Electronic Technology
	University of Windsor, Windsor, Ontario 1998 – 1999 M.Sc. Mathematics