

RANDOM METHODS FOR IDENTIFYING  
NONREDUNDANT CONSTRAINTS

by

Robert L. Smith<sup>1</sup>  
Jan Telgen<sup>2</sup>

Technical Report No. 81-4

1. Department of Industrial and Operations Engineering  
The University of Michigan, Ann Arbor, Michigan,  
48109.
2. Center for Operations Research and Econometrics,  
1348 Louvain-la-Neuve, Belgium.

April 1981



RANDOM METHODS FOR IDENTIFYING  
NONREDUNDANT CONSTRAINTS

by

Robert L. Smith and Jan Telgen

Abstract

A method to generate uniformly distributed points in a bounded region is used to develop a class of random methods to identify redundancy in systems of linear constraints. Non-redundant constraints are identified directly and the remaining constraints are labeled as redundant. Bounds on the probability that constraints are incorrectly labeled as redundant are derived. As a result, a formula is constructed giving the number of iterations of the method necessary to guarantee that within a given probability all constraints will be correctly labeled. Experimental results with some computationally attractive variants of this class of methods are presented.



Random Methods for Identifying  
Nonredundant Constraints

by

Robert L. Smith and Jan Telgen

1. INTRODUCTION

Various methods to identify (**non**)redundant constraints in systems of linear (in)equalities have been suggested in the literature. Most of these are somehow based upon the simplex method, usually via the minimization of slack variables (e.g., Thompson, Tonge and Zionts [1966], Lisy [1971], Gal [1975, 1978], Telgen [1979]). Other methods invoke the simplex method to enumerate all extreme points of the set of feasible solutions (e.g., Balinsky [1961], Mattheis [1973]), and use this information to identify redundant constraints.

Since all methods mentioned above require the complete updated simplex tableau at every iteration, their usefulness is limited to very small problems. Therefore preprocessors have been developed that work with the original problem data (e.g., Brearly, Mitra and Williams [1975], Bradley, Brown and Graves [1980]). One drawback of these preprocessors (e.g., the REDUCE option in many commercial mathematical programming packages) is that they generally do not identify all redundancy since they are restricted to very simple operations.

Here we consider methods to identify redundant constraints based on procedures to generate random interior points in a bounded region. Basically, from such an interior point the methods search in a randomly generated direction. The constraint hit first can be shown to be nonredundant. After a great number of trials (i.e., random directions) from different interior points, the constraints that have not been hit are labeled as redundant.

The first such method was suggested in Boneh and Golan [1979]. Independently Smith [1980] developed part of the same method as a procedure to generate uniformly distributed points in a bounded region. In this paper we build upon the latter results to give a class of random methods to identify non-redundant constraints.

Section 2 contains a statement of the general method and the theorems required to establish its validity. Also in that section we prove that the hitpoints are uniformly distributed over the boundary of the polytope. This result enables us to conclude that the probability of hitting a given constraint is proportional to the area of the corresponding facet of the feasible polytope. As a consequence we construct stopping criteria for the method, guaranteeing (with a prespecified probability) identification of all nonredundant constraints with facets of a minimum prespecified size.

In Section 3 we give some experimental results for computationally attractive variants of the method, such as

the one in which only coordinate directions are taken. The test problems are taken from Karwan, Telgen and Zionts [1981], an extensive computational study on redundancy identification, where results for other methods are readily available.

The random methods perform relatively well; in only a fraction of the time required by the fastest of the other methods, our method identified more than (80%) of all nonredundant constraints. Moreover its relative performance is much better on larger problems.

In the conclusion we discuss possible extensions of this approach to other areas (e.g., nonlinear systems) and to other problems (e.g., optimization).

### 3. THE ALGORITHM AND ITS PROBABILISTIC ANALYSIS

We start with a feasible region  $S$  defined by a system of inequality constraints

$$(2.1) \quad g_i(x) \leq b_i, \quad i = 1, 2, \dots, m$$

with  $x \in \mathbb{R}^n$  and  $b_i \in \mathbb{R}$  for all  $i$ . We could require a regularity condition on the  $g_i$ 's so that the region  $S$  had content and surface area. However, to simplify the exposition we will restrict discussion here to the case where  $S$  is defined by a system of linear inequalities

$$(2.2) \quad Ax \leq b$$

with  $A \in R^{m \times n}$ ,  $b \in R^m$ , and  $x \in R^n$ . Rows of  $A$  will be denoted as  $a_i^T$ ,  $i = 1, 2, \dots, m$  and without loss of generality we will assume that  $\|a_i\| = 1$ . We define the feasible region

$$(2.3) \quad S = \{x \in R^n \mid Ax \leq b\}.$$

A point  $x^0 \in R^n$  is called an interior point of  $S$  if  $Ax^0 < b$ . It is assumed that such a point exists; if some of the constraints form implicit equalities then a point in the relative interior of  $S$  will suffice.

Now the algorithm consists of three parts: the generation of interior points of  $S$  (Steps 1 and 3), the identification of nonredundant constraints (Step 2), and a stopping criterion (Step 4).

The algorithm proceeds by passing lines through interior points and identifying constraints struck by the lines as nonredundant. The points of intersection of the lines with the boundary of  $S$  are called hit points. The lines generated serve the dual purpose of providing the next interior iteration points.

Intialization: Find an interior point:  $x^0$ . Set  $k = 0$ .

Step 1: Generate a random Direction  $D^k$  uniformly distributed over a Direction Set D (a centrally symmetric subset of the boundary of a unit hypersphere).



Step 2: Determine:

$$(2.4) \quad \lambda_i = \frac{b_i - a_i^T X^k}{a_i^T D_k} \quad i = 1, 2, \dots, m$$

$$(2.5) \quad \lambda^+ = \min_i \{\lambda_i \mid \lambda_i > 0\} = \lambda_{i^+}$$

$$(2.6) \quad \lambda^- = \max_i \{\lambda_i \mid \lambda_i < 0\} = \lambda_{i^-}$$

Identify constraints  $i^+$  and  $i^-$  as nonredundant

Step 3: Generate  $U \in R$  from a uniform distribution on the interval  $(0, 1)$  and set:

$$X^{k+1} = X^k + (\lambda^- + U(\lambda^+ - \lambda^-)) D^k.$$

Step 4: If a stopping criterion is satisfied: stop; otherwise set  $k = k + 1$  and proceed with Step 1.

Several comments are in order. (1) Determining a point uniformly distributed on a unit hypersphere (or a part thereof) in Step 1 may be accomplished by setting  $D^k = N / \|N\|$  where  $N$  is a  $n$ -component vector of standard normal random variables (Knuth [1969]). (2) The proof that the constraints identified in Step 2 are nonredundant is very simple and therefore omitted (see also Boneh and Golan [1979]). In fact it can be shown more generally that the constraint closest to an iteration point of  $S$  is nonredundant; then by considering the hit points as

limiting cases of iteration points the result used here follows immediately. Obviously uniqueness of the minima in (2.5) and (2.6) is required but that is true with probability one for the points generated by the random method. (3) The points  $x^0, x^1, x^2, \dots$ , correspond to the iteration points, while  $Y_+^k = X^k + \lambda^+ D^k$  and  $Y_-^k = X^k + \lambda^- D^k$  correspond to the hit points.

Different choices for the direction set  $D$  lead to different versions of the algorithm. The version of Boneh and Golan [1979] corresponds to  $D$  equal to the entire hypersphere. Several versions are explored as to their relative computational merits in Section 3. In any case, so long as  $D$  is centrally symmetric (i.e.,  $-d \in D$ , whenever  $d \in D$ ), then the iteration points are asymptotically uniformly distributed within  $S$  [Smith [1981]].

More important for our purposes however is that the hit points are asymptotically uniformly distributed over the boundary of  $S$ . A case where this is directly evident is where we restrict iteration points to a  $\epsilon$ -strip around the boundary of  $S$ . Letting  $\epsilon \rightarrow 0$ , the iteration and hit points coincide and the uniformity of the hit points over the boundary follows. This algorithm informally referred to as "Shake and Bake" incidentally may have considerable computational merit. The result below will serve as the basis for establishing stopping criteria for the general class of random algorithms.

THEOREM 1: The hit points  $Y_+^k$  and  $Y_-^k$  of the Algorithm are asymptotically uniformly distributed over the boundary of  $S$ .

PROOF: We shall argue the result separately for the two sequences  $\langle Y_+^k \rangle$  and  $\langle Y_-^k \rangle$ . Without loss of generality, consider the sequence  $Y^k = Y_+^k, Y^0, Y^1, Y^2, \dots$  is a continuous state Markov chain and by a result in Smith [1981] it suffices to show that the transition probability density function  $f(y_1, y_2) > 0$  is symmetric, i.e., that  $f(p, q) = f(q, p)$  for all  $q, p$  on the boundary of  $S$ . Let  $W, U,$  and  $V$  be the iteration points giving rise to the two successive hit points  $p$  and  $q$  (see Figure 1). Now we go from hit point  $p$  to hit point  $q$  (in that order) iff we go from iteration point  $W$  to iteration point  $U$  to iteration point  $V$  (in that order). But the order  $W$  to  $U$  to  $V$  is just as likely as the order  $V$  to  $U$  to  $W$  since the direction set  $D$  is symmetric. Hence the likelihood of going from  $p$  to  $q$  is the same as the likelihood of going from  $q$  to  $p$ , and the result follows. ■

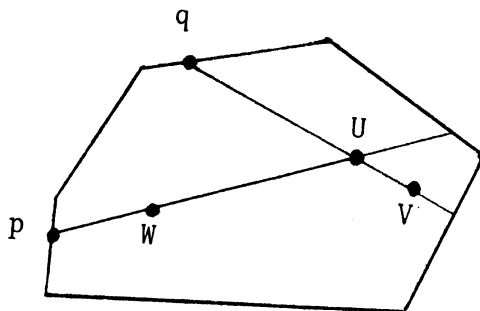


Figure 1

It is immediate from Theorem 1 that the probability  $p_i$  of identifying constraint  $i$  as nonredundant during a given iteration is proportional to the surface area  $\alpha_i$  of that facet (this presumes that the algorithm has iterated long enough to achieve uniformity). In fact, we have

$$(2.7) \quad p_i = \frac{\alpha_i}{\sum_{i=1}^l \alpha_i} \quad i = 1, 2, \dots, l$$

where there are  $\ell \leq m$  nonredundant constraints in all. It is argued in Smith [1981] that a random permutation of the indices of the points generated  $Y^{(0)}, Y^{(1)}, Y^{(2)}, \dots, Y^{(s)}$  for a large sample size  $s$  will behave as if they were independent as well as identically distributed uniform. Since our algorithm for nonredundant constraint identification is permutation invariant, we may model the hit points  $Y^k$  for large  $s$  as i.i.d. random variables. In particular, the events corresponding to constraints identified as nonredundant are modeled as independent over the iterations.

Let  $K$  be the number of independent and uniformly distributed hit points  $Y^k$  over the boundary of  $S$  necessary to identify all  $\ell$  facets (nonredundant constraints) of  $S$ . (Since the two hit points  $Y_+^k$  and  $Y_-^k$  are strongly correlated, we will be conservative and identify only one hit point per iteration).

Theorem 2: The cumulative distribution function of  $K$  is

$$(2.8) \quad F(k) = P(K \leq k) = 1 - \sum_{i=1}^{\ell} (1-p_i)^k + \sum_{i < j} (1-p_i-p_j)^k \\ - \sum_{i < j < h} (1-p_i-p_j-p_h)^k + \dots + (1-p_1 - p_2 - \dots - p_{\ell})^k$$

(Note: the last term is zero) and the expected number of iterations  $E(K)$  necessary to identify all facets is:

$$(2.9) \quad E(K) = \sum_{i=1}^{\ell} (p_i)^{-1} - \sum_{i<j} (p_i + p_j)^{-1} \\ + \sum_{i<j<h} (p_i + p_j + p_h)^{-1} - \dots \pm 0.$$

Proof: Let  $q_i$  be the number of points that identify facet  $i$  out of a total of  $k$  points. Then  $K > k$  if and only if  $q_1 = 0$  or  $q_2 = 0$  or ...  $q_{\ell} = 0$  thus:

$$p(K > k) = p(q_1 = 0 \text{ or } q_2 = 0 \text{ or } \dots \text{ or } q_{\ell} = 0)$$

$$\text{Let } \tilde{p}_i = p(q_i = 0) = (1-p_i)^k$$

$$\tilde{p}_{ij} = p(q_i = 0 \text{ and } q_j = 0) = (1-p_i-p_j)^k$$

$$\tilde{p}_{ijh} = p(q_i = 0 \text{ and } q_j = 0 \text{ and } q_h = 0) = (1-p_i-p_j-p_h)^k$$

etc.

$$\text{Let } Q_1 = \sum_{i=1}^{\ell} \tilde{p}_i$$

$$Q_2 = \sum_{i<j} \tilde{p}_{ij}$$

$$Q_3 = \sum_{i<j<h} \tilde{p}_{ijh}$$

Note that  $Q_1$  has  $\ell$  terms in all,  $Q_2$  has  $\binom{\ell}{2}$  terms  $Q_3$  has  $\binom{\ell}{3}$  terms, etc.

Then from the exclusion - inclusion theorem (see Feller [1968] Vol. 1, p. 99) we get

$$P(q_1 = 0 \text{ or } q_2 = 0 \text{ or } \dots \text{ or } q_\ell = 0) = Q_1 - Q_2 + Q_3 - \dots \pm Q_\ell.$$

From

$$F(k) = 1 - Q_1 + Q_2 - Q_3 + \dots \pm Q_\ell$$

our first result (2.8) follows.

As for the second result

$$\begin{aligned} E(K) &= \sum_{k=0}^{\infty} P(K > k) \\ &= \sum_{k=0}^{\infty} \left[ \sum_{i=1}^m (1-p_i)^k - \sum_{i < j} (1-p_i-p_j)^k + \sum_{i < j < h} (1-p_i-p_j-p_h)^k \right. \\ &\quad \left. - \dots \pm (1-p_1 - p_2 - \dots - p_\ell)^k \right]. \end{aligned}$$

Interchanging the order of summation, which is justified since all terms are nonnegative, yields

$$\begin{aligned} E(K) &= \sum_{i=1}^m \sum_{k=0}^{\infty} (1-p_i)^k - \sum_{i < j} \sum_{k=0}^{\infty} (1-p_i-p_j)^k \\ &\quad + \sum_{i < j < h} \sum_{k=0}^{\infty} (1-p_i-p_j-p_h)^k - \dots \\ &\quad \pm \sum_{k=0}^{\infty} (1-p_1 - p_2 - \dots - p_\ell)^k. \end{aligned}$$

The result then follows from the fact that

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} \quad \text{for } |x| < 1. \quad \blacksquare$$

Theorem 3:

$$(2.10) \quad E(K) \leq \sum_{i=1}^m 1/p_i$$

Proof: Since the negative terms in (2.9) are larger than the next positive terms, we get an upper bound on  $E(K)$  by neglecting the second and all succeeding terms. Hence (2.10) follows.  $\blacksquare$

Note: this bound could be obtained more directly by considering  $E(K)$  for a specific sequence of identified facets, say  $i = 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow \ell$ . Then  $E(K)$  is the sum of  $\ell$  (independent) geometric random variables and the result follows.

We can also provide a lower bound on  $E(K)$  that is more restrictive than the obvious lower bound of  $\ell$ .

Theorem 4:

$$(2.11) \quad E(K) \geq \ell \sum_{i=1}^{\ell} \frac{i}{\ell} \approx \ell \ln \ell \quad \text{for large } \ell$$

Proof: First observe from (2.9) that  $E(K)$  will attain its minimal value if all  $p_i$  are equal, i.e.,  $p_i = \frac{1}{\ell}$   $i = 1, 2, \dots, \ell$ . We are then reduced to the so-called (coupon) Collector's Problem. See Feller [1968], Vol. 1, p. 225 for a proof that  $E(K)$  takes on the form stated.  $\blacksquare$

Using Theorem 2 and with knowledge of  $p_1, p_2, \dots, p_\ell$ , we could establish a minimum number of iterations  $k$  necessary to identify all nonredundant constraints with probability  $1 - \alpha$ . However, neither the  $p_i$  nor even  $\ell$  is typically known. An alternative is establishing such a  $k$  when all  $p_i \geq p_{\min}$  where  $p_{\min}$  is prespecified and small. The following lemma is a first step toward that goal.

Lemma 5:

$$\underline{F}(k) = 1 - \sum_{i=1}^{\ell} (1-p_i)^k \leq F(k) \leq 1 - \sum_{i=1}^{\ell} (1-p_i)^k + \sum_{i<j} (1-p_i-p_j)^k$$

$= \bar{F}(k)$ . The error  $\epsilon$  involved in estimating  $F(k)$  by  $\underline{F}(k)$  is at most

$$\epsilon < \binom{\ell}{2} (1-2p_{\min})^k.$$

Proof: The first line of inequalities follows from the Bonferroni Inequalities (Feller [1968], p. 110). The bound for  $\epsilon = F(k) - \underline{F}(k) \leq \bar{F}(k) - \underline{F}(k) = \sum_{i<j} (1-p_i-p_j)^k \leq \binom{\ell}{2} (1-2p_{\min})^k$ .

The error involved in the estimate  $F(k) = \underline{F}(k)$  is by Lemma 5 negligibly small in most cases. For example, for  $\ell = 100$  constraints and  $p_{\min} = 0.01$ ,  $\epsilon \leq 0.001$  for  $k \geq 750$ .

We shall employ this approximation in the theorem below.

Theorem 6: Let  $k^*$  be the number of iterations necessary to identify all  $\ell$  nonredundant constraints with probability  $1 - \alpha$ . Set  $r = (\frac{1}{\ell})/p_{\min}$ . Then, for large  $r$ ,



$$k^* \leq \gamma(r, \alpha, \ell) \ell \ln \ell$$

where  $\gamma(r, \alpha, \ell) = r(1 + \ln \frac{1}{\alpha} / \ln \ell)$ .

PROOF: We want  $F(k) \geq 1 - \sum_{i=1}^{\ell} (1-p_i)^k \geq 1 - \ell(1 - 1/(r\ell))^k \geq 1 - \alpha$ .

Hence  $k^* \leq \ln(\alpha/\ell)/\ln(1 - 1/(r\ell)) \approx \ln(\alpha/\ell)/(-1/(r\ell))$  for  $r$  large.

Therefore  $k^* \leq r\ell(\ln \ell - \ln \alpha) = r(1 + \ln(1/\alpha)/\ln \ell)\ell \ln \ell$ . ■

Theorem 6 provides a stopping rule on the number of iterations to perform  $k^*$ . Note that  $\gamma(r, \alpha, \ell)$  is the inflation factor of the number of iterations required for a regular polytope case of  $p_i = 1/\ell$  for all  $i = 1, 2, \dots, \ell$  which we know to vary as  $\ell \ln \ell$ . Note also that  $\gamma(r, \alpha, \ell) \approx r$  for large  $\ell$ ; that is, if the smallest facet area is  $\frac{1}{r}$ th of the average for a regular polytope, then the number of iterations required is  $r$  times as great. The order of magnitude is however still  $\ell \ln \ell$ . As an example, for  $\ell = 100$  nonredundant constraints,  $r = 10$ , and  $\alpha = 0.05$ , we get  $k^* \leq 7600$  iterations. Finally, we should mention that the dimension of the space,  $n$  is of no influence on any of the results derived here.

#### 4. EXPERIMENTAL RESULTS

To apply the method described in the previous section we need an initial interior point in  $S$ . If not readily available, this may be obtained from a simplex Phase I procedure. Such a procedure is required as well if there are (explicit or implicit) equality constraints in the system.

The experimental results given below are taken from Karwan, Telgen and Zionts [1981]. The tests involved both randomly generated problems and structured problems taken from the literature. The random testproblems were generated using a modified version of LPGENR (Michaels and O'Neill [1975]) according to the following table.

Set	No. of Variables	No. of Constraints	Degeneracy	Planned Redundancy
1	10	20	None	None
2	10	20	50%	None
3	10	30	None	10 Constraints
4	10	30	50%	10 Constraints
5	10	30	None	None
6	10	30	50%	None

Table 1: Randomly Generated Testproblems

- Notes:
- 50% degeneracy means that in two different extreme points only 5 variables have a nonzero value.
  - sets 3 and 4 were obtained from sets 1 and 2 by the addition of linear combinations of constraints.
  - each set contains 5 problems.

The characteristics of the structured problems are summarized in Table 2.

Problem	No. of Variables	No. of Constraints	Source
A	12	26	Disbursement problem, Nykamp and Spronk [1978]
B	10	22	Disbursement problem, Nykamp and Spronk [1978]
C	20	29	Diet problem, Dantzig [1963]
F	15	37	Production planning Meyerman [1979]
G	23	59	Production planning Meyerman [1956]
I	5	29	Production planning, Tischer [1968]

Table 2: Structured Testproblems

Admittedly these testproblems are very small, but most redundancy identification methods are not efficient on much larger problems. In Karwan, Telgen and Zionts [1981] it is concluded that the method by Lotfi [1981] using components from the methods by Zionts, Gal and Telgen performs best in identifying all redundant (and nonredundant) constraints. We will use this method to compare with the following variants of the random method developed in the preceding section:

- (a) "basic version": purely random directions (D equal to a unit hypersphere) determination of hit points along generated direction only.

- (b) "simple coordinate version": random directions parallel to coordinate axes (D equal to the intersection of a unit hypersphere with the coordinate axes), determination of hit points along generated direction only.
- (c) "multiple coordinate version": random directions parallel to coordinate axes, determination of hit points in all coordinate directions.

We started the random methods from an interior point obtained from perturbing the starting point of Lotfi's procedure (a feasible extreme point of S). In neither of the tables below where the results are summarized do we take into account the time and effort to obtain a starting point.

Problem Set	LOTFI'S METHOD		RANDOM METHOD					
	Actual No. Nonred. Constr.	Secs. of CPU Time	Variant (a)		Variant (b)		Variant (c)	
			No. Nonred. Constr.	No. Iter. 20 Sec.	No. Nonred. Constr.	No. Iter. in 1 Sec.	No. Nonred. Constr.	No. Iter. in 1 Sec.
1	19.4	.146	13.8	566.4	17.6	814.0	17.4	100.2
2	16.4	.240	11.0	567.2	14.8	556.0	14.8	116.6
3	19.4	.418	12.2	381.8	17.4	554.0	17.0	81.0
4	16.4	.578	10.8	375.0	14.8	566.0	14.4	79.4
5	25.4	.715	10.8	376.4	21.0	568.0	19.8	82.4
6	19.8	1.250	11.8	376.0	15.8	620.4	16.0	82.2

Table 3: Performance on the randomly generated test problems (averaged over 5 problems per set)

Problem	Actual No. of Nonred. Constraints	LOTFI'S METHOD Secs. of CPU Time	RANDOM METHOD	
			Variant (a) No. of Nonred. Constraints in 20 Secs.	Variant (c) No. of Nonred. Constraints in 1 Sec.
A	13	.13	12 (92%)	13 (100%)
B	11	.11	11 (100%)	11 (100%)
C	29	.20	20 (69%)	23 (79%)
F	23	.47	16 (70%)	20 (87%)
G	54	2.26	Not Available	41 (76%)
I	11	.37	Not Available	11 (100%)

Table 4: Performance on the Structured Problems

In addition to the results mentioned in Tables 3 and 4 we found that most of the nonredundancy identified by the random method was identified in the earlier stages of applying the method. As typical, we mention the percentages of nonredundancy found in the first quartile (.25 seconds) of applying the random method to the randomly generated test problems: for variants (a), (b) and (c) these were 88.6%, 96.0% and 96.8% respectively (averaged over 30 problems). Also the last nonredundant constraint identified on the structured problems was identified after only .11 seconds on the average.

In evaluating these results it is immediately clear that variant (a) is dominated by variants (b) and (c) as a consequence of the computational savings achieved by using coordinate directions. (See also Karwan, Telgen and Zionts [1981]). Variant (b) performed slightly better than variant (c), which may be caused by the generation of more iteration points in variant (b).

In comparing the random method with Lotfi's method on the randomly generated problems, we see that at least 80% of all nonredundant constraints were identified in 0.25 seconds CPU time, whereas Lotfi's method took an average of 0.56 seconds to find all nonredundant constraints. On the structured problems the random method took an average of 0.11 seconds to identify 90% of all nonredundant constraints, whereas Lotfi's method took an average of 0.58 seconds. In conclusion therefore it is possible to identify a large part of the nonredundancy

(say 80%) in a small amount of time (20% of that required to find all redundancy).

In these experiments we did not use any of the stopping criteria developed in Section 2. It is interesting to note however what their application would have implied. Since these stopping criteria require input on uncertain parameters, it is important to note the influence of approximating certain values. Therefore we used Theorem 6 to construct the following table of upper bounds on  $k^*$  (the number of iterations required to identify all nonredundant constraints with probability  $1 - \alpha$ ) for  $\alpha = .05$ . To illustrate these bounds we used the characteristics of the randomly generated testproblems.

Problem Set	r = 5		r = 10	
	$\ell = m$	True $\ell$	$\ell = m$	True $\ell$
1	599	564	1198	1129
2	599	461	1198	923
3	960	564	1919	1129
4	960	461	1919	923
5	960	777	1919	1554
6	960	599	1919	1198

Table 5: Bounds on  $k^*$



From Table 5 it is seen that it is relatively easy to find all facets with a "large" area (small  $r$ ). Also the estimated number of nonredundant constraints should not be too far away from the true value of  $\ell$  (as in sets 3 and 4) since that induces too many unnecessary iterations.

Finally note that it is possible to derive the magnitude of the facet areas that have been hit with probability  $1 - \alpha$  after  $k$  iterations:

$$r \geq \frac{k}{\ell \ln \frac{\ell}{\alpha}}$$

For example if  $k = 360$ ,  $\alpha = .05$ , and  $\ell = 20$ , then  $r \geq 3$ .

## 5. CONCLUSION

Thus far the random methods have been viewed as preprocessors that are capable of identifying nonredundant constraints. The nonidentified constraints can then be omitted in an optimization algorithm although the solution has to be checked for feasibility afterwards, since the omission of some constraints may not be justified. As an aid in this process we have developed stopping criteria and derived probability statements for the random methods. This approach can be extended further by the inclusion of Bayesian arguments; we hope to report on this in the near future.

In addition to an application as a preprocessor, the random methods could also be used as integral parts of an

optimization procedure, especially in the Bayesian mode approach. This topic is currently under investigation.

Although we described the random methods entirely in terms of linear constraints, they are applicable to nonlinear constraints as well. Boneh and Golan [1979] give an example of a problem with quadratic constraints that could be handled only after removal of the redundant constraints. Furthermore, with some minor modifications, nonconvex feasible regions (and even disconnected ones) can be treated by the same class of random methods.

## REFERENCES

1. Balinsky, M. L. [1961], "An Algorithm for Finding all Vertices of Convex Polyhedral Set," J. Soc. Ind. Appl. Math 9 (1), 72-88.
2. Boneh, A. and Golan, A., [1979], "Constraint Redundancy and Feasible Points Generator," presented at EURO III, Amsterdam, also in Karwan, Telgen and Zionts [1981].
3. Bradley, G. H., Brown, G. G. and Graves, G. W. [1980], "Structural Redundancy in Large-Scale Optimization Models," Technical Report, Naval Postgraduate School, Monterey, California 93940: also in Karwan, Telgen and Zionts [1981].
4. Brearley, A. L., Mitra, G., and Williams, H. P. [1975], "Analysis of Mathematical Programming Problems Prior to Applying the Simplex Algorithm," Math. Progr. 8, 54-83.
5. Dantzig, G. B. [1963], Linear Programming and Extensions, Princeton University Press.
6. Feller, W. [1968], An Introduction to Probability Theory and its Applications, Vol. I, John Wiley & Sons, Inc.
7. Gal, T. [1975], "Zur Identifikation Redundanter Nebenbedingungen in Linearen Programmen," Reitschrift Fur Operations Research 19, 19-28.
8. Gal, T., [1978], Redundancy in Systems of Linear Inequalities Revisited, Discussion Paper 19, Fern Universität, Magen BRD.
9. Karwan, M. H., Telgen, J. and Zionts, S. [1981], Redundancy in Mathematical Programming, Springer Verlag, New York-Heidelberg-Berlin.
10. Knuth, D. E. [1969], The Art of Computer Programming, Addison-Wesley, Reading, Mass.
11. Lisý, J. [1971], "Metody Pro Nalezini Redundantnich Omereni v Ulohach Linearniho Programovani," Ekonomicho Mathematicky Obror, 7 (3), 285-298.
12. Lotfi, V. [1981], "Redundancy in Mathematical Programming," Dissertation, State University of New York, Buffalo, New York 14260.
13. Mattheis, T. H. [1973], "An Algorithm for Determining Irrelevant Constraints and all Vertices in Systems of Linear Inequalities," Operations Research 21, 247-260.



14. Meyerman, G. L. [1966], "Betekenis Van Een Aantal Cultuurtechnische Jakboren Voor De Ontwikkelings Mogelykheden - Van Veenkoloniale Akkerbouw Bedryven Dissertation, Agricultural University Wageningen, The Netherlands.
15. Meyerman, B. G. [1979], "Private Communication."
16. Michaels, W. M. and O'Neill, R. P. [1975], "User's Guide for LPGENR," Dept. of Computer Science, Louisiana State University, Baton Rouge, LA.
17. Nykamp, P. and Spronk, J. [1978], "Three Cases in Multiple Criteria Decision Making, an Interactive Multiple Goal Programming Approach," report 7822, Centre for Research in Business Economics, Erasmus University, Rotterdam.
18. Smith, R. L. [1981], "Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions," report 81-1, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.
19. Telgen, J. [1979], Redundancy and Linear Programs, Mathematical Center, Amsterdam.
20. Thompson, G. L., Tonge, F. M. and Zionts, S. [1966], "Techniques for Removing Non-binding Constraints, and Extraneous Variables from Linear Programming Problems," Management Science 12 (7) 588-608.
21. Tischer, H. J. [1968], "Mathematische Verfahren zur Redusierung der Seiden - Und Spaltenzahl Linearer Optimierungsaufgaben," Dissertation, Central Institut fur Fertigungstechnik des Maschivenbaues, Karl Marx Stadt, DDR.