SHAKE-AND-BAKE ALGORITHMS FOR GENERATING UNIFORM POINTS ON THE BOUNDARY OF BOUNDED POLYHEDRA

C. G. E. BOENDER
Erasmus University, Rotterdam, The Netherlands

R. J. CARON and J. F. McDONALD
University of Windsor, Windsor, Ontario, Canada

A. H. G. RINNOOY KAN and H. E. ROMEIJN
Erasmus University, Rotterdam, The Netherlands

R. L. SMITH
University of Michigan, Ann Arbor, Michigan

J. TELGEN
University of Twente, Enschede, The Netherlands

A. C. F. VORST
Erasmus University, Rotterdam, The Netherlands

(Received March 1989; revisions received February, June 1990; accepted July 1990)

We present a class of shake-and-bake algorithms for generating (asymptotically) uniform points on the boundary of full-dimensional bounded polyhedra. We also report results of simulations for some elementary test problems.

In this paper, we study the so-called *shake-and-bake* algorithms which, as far as we know, provide the only practical way of generating (asymptotically) uniform points on the boundary ∂S of a fully-dimensional polytope S. The fact that the points are uniformly distributed means that, in the long run, all subsets of ∂S of equal size will be visited equally often, and subsets with positive measure will be visited with probability one. Part of the material that we present in this paper may be found in more detail in the technical reports of Boender et al. (1988a, b). For the so-called hit-and-run algorithms, which generate asymptotically uniform points on the interior of S, we refer to Smith (1984) and Berbee et al. (1987).

The relevance of generating points on ∂S, and the name shake-and-bake for this class of algorithms were mentioned in Smith and Telgen (1981) in the context of detecting necessary constraints. Note that the shake-and-bake algorithms can also be used for the problem of optimizing functions which attain their optimal value on ∂S, such as, for example, the minimization of a concave function over S (see, e.g., Patel and Smith 1983).

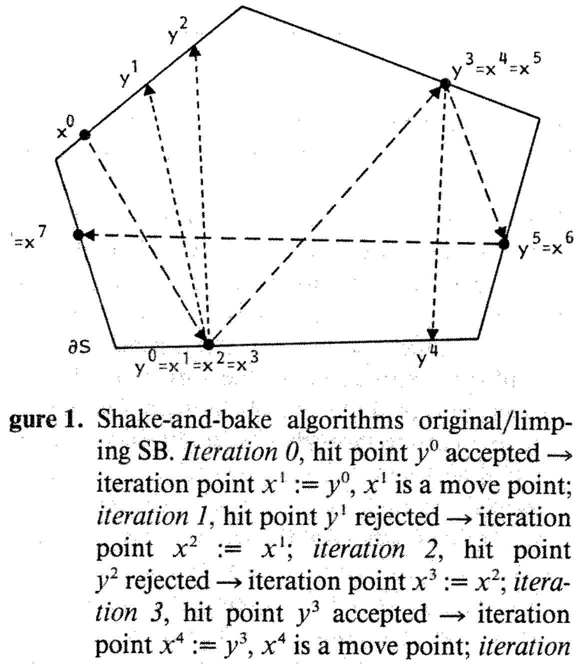
Smith (1982) suggested the first shake-and-bake algorithm, referred to here as *original SB*, that generates a sequence of points which are asymptotically uniformly distributed on ∂S. Given an *iteration point* $x^0 \in \partial S$, a random search vector v is generated from the uniform distribution on the surface of the unit hypersphere with center x^0. The intersection point y^0 of the line passing through x^0 with direction vector v with ∂S is accepted as a *move point* (i.e., $x^1 = y^0$) with probability $\cos \phi_{x^0}/(\cos \phi_{x^0} + \cos \phi_{y^0})$, where ϕ_{x^0} and ϕ_{y^0} are the acute angles of the search vector v with the normals to ∂S at x^0 and y^0, respectively; else $x^1 = x^0$. The intersection point y^0 is referred to as *hit point*. Hence, if the hit point y^0 is accepted as a move point, then the next iteration point x^1 is equal to y^0, else the next iteration point x^1 is equal to the current one x^0.

Subject classification: Simulation, random variable generation: generating asymptotically uniform points on the boundary of a polytope.

0030-364X/91/3906-0945 $01.25

In the next SB algorithm, *limping SB*, the *move robability* is equal to cos ϕ_{x^0}, while the search vector is drawn from the same distribution as for original B. The limiting distribution of the sequence of iter-tion points, as well as the sequence of move points, enerated by limping SB is proven to be uniform on S. The third algorithm, *running SB*, chooses the earch vector from a distribution such that the hit oint is *always* accepted as a move point, while main-ining the important property that the distribution of e iteration points is asymptotically uniform on ∂S. See Figures 1 and 2.)

In this paper, we give a proof for the uniform miting distribution of the iteration points which pplies to a large class of shake-and-bake algorithms, ncluding the three algorithms just mentioned. The utline of this paper is as follows: In Section 1, we will escribe the general class of shake-and-bake algo-thms and discuss the above special cases in more etail. Section 2 contains a proof that the limiting istribution of the sequence of iteration points gener-ed by the algorithms is uniform on the boundary of . In Section 3 we present some experimental results.



gure 1. Shake-and-bake algorithms original/limping SB. *Iteration 0*, hit point y^0 accepted → iteration point $x^1 := y^0$, x^1 is a move point; *iteration 1*, hit point y^1 rejected → iteration point $x^2 := x^1$; *iteration 2*, hit point y^2 rejected → iteration point $x^3 := x^2$; *iteration 3*, hit point y^3 accepted → iteration point $x^4 := y^3$, x^4 is a move point; *iteration 4*, hit point y^4 rejected → iteration point $x^5 := x^4$; *iteration 5*, hit point y^5 accepted → iteration point $x^6 := y^5$, x^6 is a move point; and *iteration 6*, hit point y^6 accepted → iteration point $x^7 := y^6$, x^7 is a move point.

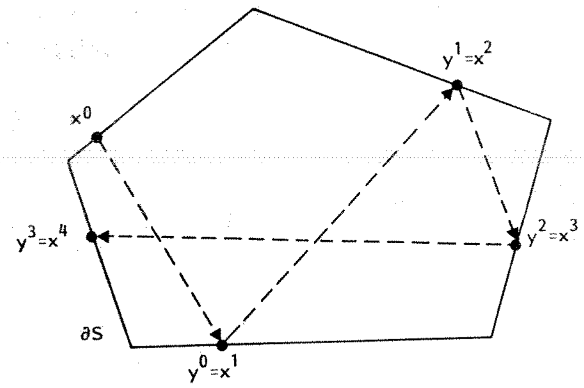


Figure 2. Shake-and-bake algorithm running SB.

1. THE SHAKE-AND-BAKE ALGORITHMS

1.1. Introduction

Consider a feasible region $S \subset \mathbb{R}^d$ $(d \geq 2)$ defined by the system of linear inequalities:

$$a_i' w \leq b_i \quad (i = 1, \ldots, m) \tag{1}$$

where we have normalized the coefficients of the inequalities by choosing $\| a_i \| = 1$. Assume that S is bounded, nonempty and of full dimension. Then S is a polytope that contains interior points, i.e., points for which the inequalities (1) are all satisfied as strict inequalities. We assume, without loss of generality, that all of the constraints are nonredundant, i.e., none of them can be dropped from the system (1) without changing S.

Let ∂S^0 be the set of points in S for which exactly one constraint is binding, that is,

$$\partial S^0 = \bigcup_{i=1}^{m} \{w : a_i' w = b_i;\ a_j' w < b_j,\ j \neq i\}.$$

The shake-and-bake algorithms are based on a search from some point $x \in \partial S^0$ in a feasible direction v, i.e., if constraint k is binding at x, then $a_k' v < 0$. The intersection point y with the constraint hit first in the direction v is referred to as a *hit point* and becomes a *move point* with probability $\beta(y \mid x)$.

The hit point y is computed as follows. Let ∂V_i be the bounding hyperplane of the half-space

$$V_i := \{w : a_i' w \leq b_i\} \quad (i = 1, \ldots, m).$$

To determine y we compute all intersection points of the straight line passing through x with direction vector v, denoted by

$$x + \lambda v \quad \lambda \in \mathbb{R}$$

with the hyperplanes ∂V_i, $(i = 1, \ldots, m)$. It is easy to show that the intersection points correspond to the following values of λ:

$$\lambda_i = \frac{b_i - a_i' x}{a_i' v} \quad (i = 1, \ldots, m).$$

Clearly, the intersection point corresponding to the smallest positive value of λ is the hit point, which, for the algorithms described below, is an element of ∂S^0 with probability one.

1.2. The Class of Shake-and-Bake Algorithms

The class of shake-and-bake algorithms for polyhedral sets can be described as follows (see Figure 1):

Step 0. Find some point $x^0 \in \partial S^0$. Define k as the index corresponding to the constraint that is binding at x^0. Set $n := 0$.

Step 1. Generate a direction vector v from an absolutely continuous probability distribution over the intersection of the half-space $\{w : a_k' w \leqslant 0\}$ with the surface of the d-dimensional unit hypersphere centered at the origin.

Step 2. Determine the hit point y^n as:

$$\lambda_i := \frac{b_i - a_i' x^n}{a_i' v} \quad (i = 1, \ldots, m)$$

$$r := \arg \min_{1 \leqslant i \leqslant m} \{\lambda_i \mid \lambda_i > 0\}$$

$$y^n := x^n + \lambda_r v.$$

Step 3. With probability $\beta(y^n \mid x^n)$, set $x^{n+1} := y^n$ and $k := r$, i.e., y^n becomes a move point. Otherwise, set $x^{n+1} := x^n$, i.e., the next iteration point is equal to the current one.

Step 4. Set $n := n + 1$ and return to *Step 1.*

Note that from the description of the algorithms we may conclude that the sequence of iteration points defines a *Markov chain* with a *stationary transition density function* and *continuous state-space* ∂S^0, which will be useful in the subsequent analysis of the algorithms.

Because of our assumptions on set S, there is a one-to-one correspondence between a feasible direction from a particular point in ∂S^0 and a hit point. We will define our class of algorithms by imposing conditions on the distribution of the search directions and on the move probability function $\beta(y \mid x)$.

Without loss of generality, we write the density of the absolutely continuous component of the distribution of hit points in the form:

$$\rho(y \mid x) = f(x, y) \frac{b_y - a_y' x}{\| x - y \|^d}$$

where $\partial V_y := \{w : a_y' w = b_y\}$ is the hyperplane that is binding at $y \in \partial S^0$. Note that $b_y - a_y' x$ is equal to the distance from x to ∂V_y. Also note that $(b_y - a_y' x)/\| x - y \| = \cos \phi_y$, where $\cos \phi_y$ is as described in the Introduction. The move probability function will be written in the form:

$$\beta(y \mid x) = g(x, y) \frac{b_x - a_x' y}{\| x - y \|}$$

where ∂V_x is the hyperplane that is binding at $x \in \partial S^0$. Denote the density of the absolutely continuous component of the distribution of the ν-step transition probability of *moving* from $x \in \partial S^0$ to a neighborhood of $y \in \partial S^0$ by $p^{(\nu)}(y \mid x)$. The one-step transition density function $p(y \mid x) = p^{(1)}(y \mid x)$ is then given by:

$$p(y \mid x) = \beta(y \mid x) \rho(y \mid x)$$

$$= h(x, y) \frac{(b_x - a_x' y)(b_y - a_y' x)}{\| x - y \|^{d+1}}$$

where

$$h(x, y) = f(x, y) g(x, y).$$

The class of shake-and-bake algorithms we discuss in this paper is defined by imposing the following conditions on the function h:

i. $h(x, y)$ is symmetric in x and y, i.e., $h(x, y) = h(y, x)$ for all $x, y \in \partial S^0$.
ii. $h(x, y)$ is uniformly bounded away from by zero, i.e., there exists a constant $\delta_h > 0$ such that $h(x, y) \geqslant \delta_h$ for all $x, y \in \partial S^0$, $x \neq y$.

We assume that $\beta(y \mid x)$ is a measurable function, and, of course, that $0 \leqslant \beta(y \mid x) \leqslant 1$ for all $x, y \in \partial S^0$. It is easy to show that the first condition implies $p^{(\nu)}(y \mid x) = p^{(\nu)}(x \mid y)$ for all $x, y \in \partial S^0$, where $\nu = 1, 2, \ldots$. The class of shake-and-bake algorithms that satisfy conditions i and ii will be referred to as *SB algorithms.*

1.3. Some Special Cases

In this section, we will describe three specific SB algorithms.

1.3.1. Original SB

Consider the hypersphere directions (HD) hit-and-run algorithm, which generates a sequence of points in the interior of a fully-dimensional bounded set (see, e.g., Berbee et al. 1987). Starting at some point x in this set, generate a direction vector v from the uniform distribution over the unit sphere. Then choose the next point y in the sequence uniformly from the intersection of the region with the line through x in the direction v. The move probability function for original SB can be motivated heuristically by considering the HD algorithm applied to a strip having small width ϵ around the boundary ∂S of a polytope. The intersection of the set with the line through x in the direction v is now the union of two line segments. The acceptance probability for the SB algorithm then corresponds in the limit as $\epsilon \to 0$ to the relative length of the line segment in the polytope not containing the point x. In particular, for original SB the direction vector v should have the uniform distribution over the unit sphere, or equivalently, the half-sphere of feasible directions. A direction vector from the uniform distribution over the unit sphere can be generated efficiently by generating a vector from the d-variate standard normal distribution, and normalizing this vector to have a norm equal to one (see e.g., Knuth, 1969). The distribution of hit points is given by

$$\rho_O(y \mid x) = \frac{2(b_y - a_y' x)}{C_d \| x - y \|^d}$$

where

$$C_d = \frac{2\pi^{d/2}}{\Gamma(d/2)}$$

is the surface area of a d-dimensional hypersphere with unit radius.

The move probability function is given by

$$\beta_O(y \mid x) = \frac{(b_x - a_x' y)}{(b_x - a_x' y) + (b_y - a_y' x)}.$$

So for this algorithm we have

$$h_O(x, y) = \frac{2 \| x - y \|}{C_d[(b_x - a_x' y) + (b_y - a_y' x)]} \geq \delta_{h_O}$$

with

$$\delta_{h_O} = \frac{1}{C_d} > 0$$

and the one-step transition density function is equal to

$$p_O(y \mid x) = \frac{2(b_x - a_x' y)(b_y - a_y' x)}{C_d \| x - y \|^d [(b_x - a_x' y) + (b_y - a_y' x)]}.$$

1.3.2. Limping SB

In this algorithm, the direction vector has the same distribution as for original SB, i.e., $\rho_L(y \mid x) = \rho_O(y \mid x)$. The move probability function for this algorithm is chosen for its computational simplicity, and with the objective of getting a symmetric transition density function. It is given by

$$\beta_L(y \mid x) = \frac{b_x - a_x' y}{\| x - y \|} = -a_x' v.$$

Note that $\beta_L(y \mid x)$ does not depend explicitly on y. Thus, the hit point need not be computed unless it is accepted as a move point. For this algorithm we have

$$h_L(x, y) = \frac{2}{C_d}$$

so that

$$\delta_{h_L} = \frac{2}{C_d} > 0$$

and

$$p_L(y \mid x) = \frac{2(b_x - a_x' y)(b_y - a_y' x)}{C_d \| x - y \|^{d+1}}$$

(see Boender et al. 1988b for the derivation of the transition density function for this algorithm).

1.3.3. Running SB

For all SB algorithms we know that the correlation between X^n and X^{n+M} goes to zero if M goes to infinity. Consequently, randomly permuting the indices in a large number of generated points will induce the same uncorrelated effect for any pair of distinct points in the shuffled sequence of points. In other words, two elements of the shuffled sequence $X^{(1)}, \ldots, X^{(M)}$ will be *asymptotically independent*. However, for original and limping SB successive iteration points are highly correlated. This is due to the fact that not every hit point is a move point. Thus, the rate of convergence of the iteration points to being independently and uniformly distributed will be slow. The running SB algorithm solves this problem by taking

$$\beta_R(y \mid x) = 1.$$

The distribution of the direction vector is then chosen with the goal of obtaining a uniform limiting distribution. In particular, the random direction vector v is obtained as follows: Draw a point u from a uniform distribution on the (relative) interior of the intersection of the d-dimensional unit hypersphere centered at the origin and the hyperplane

$\{w: a_x' w = 0\}$. The search vector v is defined as the vector with Euclidean norm 1 whose projection on the hyperplane $\{w: a_x' w = 0\}$ is equal to u, under the condition that $a_x' v < 0$. We will describe this in more detail below. This choice of distribution of the direction vector leads to the transition density function:

$$p_R(y \mid x) = \frac{(b_x - a_x' y)(b_y - a_y' x)}{B_{d-1} \, \| x - y \|^{d+1}}$$

where

$$B_d = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)}$$

is the volume of a d-dimensional hypersphere with unit radius (see again Boender et al. 1988b for the derivation of the transition density function). Note that for this algorithm

$$h_R(x, y) = \frac{1}{B_{d-1}}$$

so that

$$\delta_{h_R} = \frac{1}{B_{d-1}} > 0.$$

In this algorithm, we need to generate a point u from the uniform distribution on the (relative) interior of a $(d - 1)$-dimensional unit hypersphere contained in some hyperplane, say, $\{w: c' w = 0\}$ (with $\| c \| = 1$). This point can be obtained in the following way: First, draw a point $\tilde{u}$ from the uniform distribution on the surface of a d-dimensional unit hypersphere centered at the origin. The point $(I - cc')\tilde{u} / \| (I - cc')\tilde{u} \|$, which is the projection of $\tilde{u}$ on the hyperplane $\{w: c' w = 0\}$, rescaled to have norm equal to 1, is uniformly distributed on the surface of the $(d - 1)$-dimensional unit hypersphere centered at the origin contained in that hyperplane. This point is rescaled to have norm r, where r is chosen such that the resulting point u has the required distribution, which means that r^{d-1} has to be drawn from the uniform distribution on (0, 1]. So we have

$$u = \frac{r(I - cc')\tilde{u}}{\| (I - cc')\tilde{u} \|} = \frac{r(I - cc')\tilde{u}}{\sqrt{1 - (c'\tilde{u})^2}}$$

and the corresponding direction vector is given by:

$$v = u - \sqrt{1 - r^2}\, c$$
$$= \frac{r}{\sqrt{1 - (c'\tilde{u})^2}}\, \tilde{u} - \left(\frac{r(c'\tilde{u})}{\sqrt{1 - (c'\tilde{u})^2}} + \sqrt{1 - r^2} \right) c.$$

1.4. Limping and Running SB Re-Examined

Given that the present iteration point is x, the conditional probability that the next hit point is also a move point is given by

$$P(x) = \int_{\partial S^0} \beta(y \mid x) \rho(y \mid x) \, dy = \int_{\partial S^0} p(y \mid x) \, dy.$$

Since for all x, $p(y \mid x) > 0$ for all y in a subset of ∂S^0 of measure greater than 0, it follows that for all SB algorithms $P(x) > 0$ for all x. Since the expected value of a random variable having a geometric distribution with parameter q is equal to $(1 - q)/q$, the expected number of iterations required to generate a move point from x is equal to $1/P(x)$, so that this expected value is always finite.

For any shake-and-bake algorithm, say A, with transition density function $p(y \mid x)$ we can define another algorithm $\tilde{A}$ generating only the move points corresponding to A. (Obviously, when $P(x) = 1$ for all x, e.g., when A = running SB, the two algorithms are the same.) The transition density function $\tilde{p}(y \mid x)$ of algorithm $\tilde{A}$ can be expressed in terms of $p(y \mid x)$ and $P(x)$:

$$\tilde{p}(y \mid x) = p(y \mid x) + (1 - P(x)) p(y \mid x) + (1 - P(x))^2 p(y \mid x) + \ldots$$
$$= \sum_{i=0}^{\infty} (1 - P(x))^i p(y \mid x)$$
$$= \frac{1}{P(x)} p(y \mid x).$$

It follows that $\tilde{h}(x, y) = h(x, y)/P(x)$. Of course, if the function $\tilde{h}$ satisfies conditions i and ii in Section 1.2, then algorithm $\tilde{A}$ is an SB algorithm. If A is an SB algorithm, then so is $\tilde{A}$ if and only if $P(x)$ is independent of x.

For limping SB the move probability $P_L(x)$ is equal to:

$$P_L(x) = \frac{2B_{d-1}}{C_d}.$$

This move probability is independent of x, so the algorithm generating the move points of limping SB is also an SB algorithm. Comparing the transition density functions of limping and running SB we observe that limping $\widetilde{\text{SB}}$ = running SB (see Figures 1 and 2).

1.5. Computational Efficiency

To determine a next iteration point all shake-and-bake algorithms have to generate a search vector. If the acceptance probability function of an algorithm depends explicitly on y, then the hit point corresponding to the search vector has to be computed. If this is not the case, then the hit point only needs to be computed if it is accepted as a move point.

For the three algorithms described above, the generation of a search vector requires $O(d)$ time. Due to the $2d$ multiplications for each inequality of S, the computation of the intersection points requires $O(md)$ time. This implies that for original and running SB the computation of an iteration point requires $O(md)$ time. For limping SB the acceptance probability does not depend explicitly on y, so the expected computation time per iteration point depends on the probability that a hit point is also a move point. Since $1/P_L(x) \approx \sqrt{\pi}(d+1)/2$ for large d, we have that this probability is $O(1/\sqrt{d})$, so the expected computation time per iteration point is $O(m\sqrt{d})$.

The foregoing analysis suggests that limping SB is better than original or running SB in some sense. However, there will be a difference among SB algorithms in the rate of convergence to the uniform distribution. As noted, the convergence rate of algorithms for which not every iteration point is a move point (e.g., original and limping SB) may well be much lower than for algorithms generating only move points, such as running SB. This issue will be addressed in some more detail in the next section.

2. THE UNIFORM LIMITING DISTRIBUTION

In this section, we prove that for all SB algorithms the random sequence $\{X^n\}_{n=0}^{\infty}$ of iteration points converges to the uniform distribution on ∂S^0, independently of the starting point in the set ∂S^0. In other words, we will prove that

$$\lim_{n\to\infty} \Pr\{X^n \in A \mid X^0 = x^0\} = \frac{m_{d-1}(A)}{m_{d-1}(\partial S)}$$

for every $A \subseteq \partial S$ and every starting point $x^0 \in \partial S$, where $m_{d-1}(\cdot)$ denotes the $(d-1)$-dimensional Lebesgue measure of a set.

We use the following theorem from Doob (1953, p. 197).

Theorem 1. *If there exists a $\delta > 0$ and a $\nu \geq 1$ (possibly depending on S) such that $p^{(\nu)}(y \mid x) \geq \delta$ for all x, $y \in \partial S^0$, then there exists a stationary absolute probability distribution $P(\cdot)$ such that*

$$|\Pr\{X^n \in A \mid X^0 = x^0\} - P(A)|$$
$$\leq (1 - \delta m_{d-1}(\partial S))^{n/\nu - 1}$$

for all $A \subseteq \partial S$ and for all $x^0 \in \partial S^0$.

We can use this theorem to get the following result.

Theorem 2. *If*

a. *there exists a scalar $\delta > 0$ and a $\nu \geq 1$ such that $p^{(\nu)}(y \mid x) \geq \delta$ for all x, $y \in \partial S^0$, and*
b. *$p(y \mid x) = p(x \mid y)$ for all x, $y \in \partial S^0$,*

then the sequence $\{X^n\}_{n=0}^{\infty}$ of points has a uniform limiting distribution on ∂S^0.

Proof. It follows from Theorem 1 that a implies the existence of an asymptotic stationary distribution. Analogous to the proof given in Smith (1984, p. 1300), it follows that b implies that the uniform distribution is the unique stationary distribution.

As noted in Section 1.2, condition b is satisfied. What remains is to find a $\delta > 0$ and a $\nu \geq 1$ such that condition a is satisfied. Theorem 3 proves that a is satisfied with $\nu = 4$.

Theorem 3. *There exists a scalar $\delta > 0$ such that $p^{(4)}(y \mid x) \geq \delta$ for all x, $y \in \partial S^0$.*

The proof will be given after the following two lemmas.

Lemma 1. *There exists an $\hat{\epsilon}_0 > 0$ such that for every $x \in \partial S^0$ there exists an index j such that $b_j - a_j'x > \hat{\epsilon}_0$.*

Proof. Suppose that such an $\hat{\epsilon}_0$ does not exist. This means that for all $\epsilon > 0$ there exists an $x_\epsilon \in \partial S^0$ such that $b_j - a_j'x_\epsilon \leq \epsilon$ for all j. So for all $n = 1, 2, \ldots$, there exists an $x_n \in \partial S^0$ such that $b_j - a_j'x_n \leq 1/n$ for all j. Since ∂S is compact, we know that the sequence $\{x_n\}_{n=1}^{\infty}$ has a limiting point $x_0 \in \partial S$ for which $b_j - a_j'x_0 = 0$ for all j. Now suppose y is an interior point of S, i.e., $b_j - a_j'y > 0$ for all j. Then $x_0 + \alpha(y - x_0)$ is an interior point for every $\alpha > 0$ since $b_j - a_j'(x_0 + \alpha(y - x_0)) = \alpha(b_j - a_j'y) > 0$ for all j. This implies that S is unbounded, which is a contradition.

Lemma 2. *For every i there exists an $\hat{\epsilon}_i > 0$, such that the set $\partial V_i^{\hat{\epsilon}_i} := \{z \in \partial V_i : b_j - a_j'z > \hat{\epsilon}_i \text{ for all } j \neq i\}$ has positive $(d-1)$-dimensional Lebesgue measure. (See Figure 3).*

Proof. Choose an arbitrary constraint i. We know that constraint i is nonredundant, so there exists an x_0 for which

$$b - a_i'x_0 = \mu_0 < 0$$

$$b_j - a_j'x_0 \geq 0 \quad (j \neq i).$$

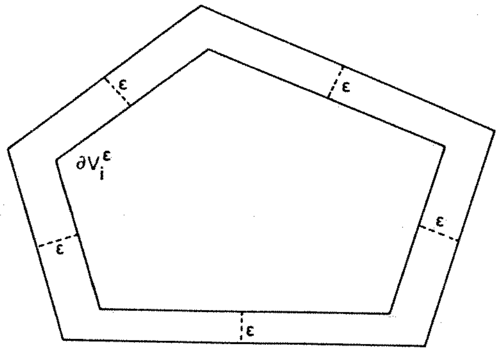


Figure 3. ∂V_i^{ϵ}.

Suppose that x_1 is an interior point of S, so

$$b_i - a_i' x_1 = \mu_1 > 0$$

$$b_j - a_j' x_1 > 0 \quad (j \neq i).$$

Choose

$$x^* = \frac{\mu_1 x_0 - \mu_0 x_1}{\mu_1 - \mu_0}$$

which is a convex combination of x_0 and x_1. Thus we have

$$b_i - a_i' x^* = b_i - \frac{\mu_0}{\mu_1 - \mu_0}(\mu_1 - b_i) + \frac{\mu_1}{\mu_1 - \mu_0}(\mu_0 - b_i) = 0$$

$$b_j - a_j' x^* > 0 \quad (j \neq i).$$

So $b_i - a_i' x^* = 0$ and $b_j - a_j' x^* \geqslant 2\hat{\epsilon}_i$ for all $j \neq i$, where

$$\hat{\epsilon}_i := \min_{j \neq i} (b_j - a_j' x^*)/2 > 0.$$

For all $z \in \partial V_i$ for which $\| z - x^* \| < \hat{\epsilon}_i$ we have

$$b_i - a_i' z = 0$$

$$b_j - a_j' z = b_j - a_j' x^* + a_j'(x^* - z) > 2\hat{\epsilon}_i - \hat{\epsilon}_i = \hat{\epsilon}_i \quad (j \neq i)$$

which implies that $z \in \partial V_i^{\hat{\epsilon}_i}$. Hence, the set $\partial V_i^{\hat{\epsilon}_i}$ has positive $(d-1)$-dimensional Lebesgue measure.

Corollary 1. *If we define $\hat{\epsilon} := \min_{i=0,1,\ldots,m} \hat{\epsilon}_i > 0$, then for all $0 < \epsilon < \hat{\epsilon}$ and for all i the set ∂V_i^{ϵ} has positive $(d-1)$-dimensional Lebesgue measure.*

Proof of Theorem 3. It is obvious that we cannot find a positive lower bound on the one-step transition density function $p(y \mid x)$ because this function is equal to zero for all points in the facet containing the point x. It can be shown that when S is a two-dimensional triangle, the two-step transition density function is everywhere positive, but has no positive lower bound. Again when S is the two-dimensional triangle, it can be shown that for running SB the three-step transition density function is not bounded from below by a positive constant, due to the fact that the acceptance probability function is always equal to 1. We will show that there exists a positive lower bound on the four-step transition density function, for all SB algorithms and for all full-dimensional polytopes. (See Figure 4.)

For all $x, y \in \partial S^0$

$$p^{(4)}(y \mid x) \geqslant \int_{\partial S^0} \int_{\partial S^0} \int_{\partial S^0} p(z_1 \mid x) p(z_2 \mid z_1) p(z_3 \mid z_2) p(y \mid z_3)\, dz_3\, dz_2\, dz_1.$$

Suppose that $x \in \partial V_i$ and $y \in \partial V_j$. Choose some $0 < \epsilon < \hat{\epsilon}$. It follows from Lemma 1 that there exists a $k \neq i$ and an $l \neq j$, such that $b_k - a_k' x > \epsilon$ and $b_l - a_l' y > \epsilon$. Choose some index $t \neq k, l$ (see Figure 4). Define r_S as the maximal distance between two points in ∂S:

$$r_S := \max_{\eta_1, \eta_2 \in \partial S} \| \eta_1 - \eta_2 \|.$$

Then:

1. $$p(z_1 \mid x) = h(x, z_1) \frac{(b_i - a_i' z_1)(b_k - a_k' x)}{\| x - z_1 \|^{d+1}} > \delta_h \frac{\epsilon^2}{r_S^{d+1}}$$

for all $z_1 \in \partial V_k$ satisfying $b_i - a_i' z_1 > \epsilon$.

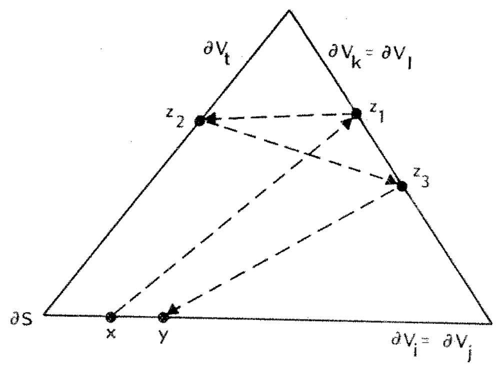


Figure 4. From x to y in four steps.

2. $p(y \mid z_3) = h(z_3, y)\dfrac{(b_l - a_l'y)(b_j - a_j'z_3)}{\| z_3 - y \|^{d+1}}$

$$> \delta_h \frac{\epsilon^2}{r_S^{d+1}}$$

for all $z_3 \in \partial V_l$ satisfying $b_j - a_j' z_3 > \epsilon$.

3. $p(z_2 \mid z_1) = h(z_1, z_2)\dfrac{(b_k - a_k' z_2)(b_t - a_t' z_1)}{\| z_1 - z_2 \|^{d+1}}$

$$> \delta_h \frac{\epsilon^2}{r_S^{d+1}}$$

for all $z_1 \in \partial V_k$ satisfying $b_t - a_t' z_1 > \epsilon$ and for all $z_2 \in \partial V_t$ satisfying $b_k - a_k' z_2 > \epsilon$.

4. $p(z_3 \mid z_2) = h(z_2, z_3)\dfrac{(b_t - a_t' z_3)(b_l - a_l' z_2)}{\| z_2 - z_3 \|^{d+1}}$

$$> \delta_h \frac{\epsilon^2}{r_S^{d+1}}$$

for all $z_2 \in \partial V_t$ satisfying $b_l - a_l' z_2 > \epsilon$ and for all $z_3 \in \partial V_l$ satisfying $b_t - a_t' z_3 > \epsilon$.

So we have

$$p^{(4)}(y \mid x) \geq \int_{\partial V_k^\epsilon} \int_{\partial V_t^\epsilon} \int_{\partial V_l^\epsilon} \delta_h^4 \frac{\epsilon^8}{r_S^{4d+4}}\, dz_3\, dz_2\, dz_1$$

$$\geq \delta_h^4 \frac{\epsilon^8}{r_S^{4d+4}} \cdot m_{d-1}(\partial V_k^\epsilon) \cdot m_{d-1}(\partial V_t^\epsilon) \cdot m_{d-1}(\partial V_l^\epsilon)$$

$$\geq \delta_h^4 \frac{\epsilon^8}{r_S^{4d+4}} \cdot \left(\min_i m_{d-1}(\partial V_i^\epsilon)\right)^3 =: \delta.$$

Using $\epsilon > 0$, $r_S < \infty$, $\delta_h > 0$, and Corollary 1 we know that $\delta > 0$.

The value of δ can be used to compare the convergence rate of the SB algorithms using Theorem 1. Clearly, a larger value for δ corresponds with a faster rate of convergence to the uniform distribution. We now rewrite the expression for δ given above as

$$\delta = \delta_h^4 \cdot \delta_S$$

where

$$\delta_S = \frac{\epsilon^8}{r_S^{4d+4}} \cdot \left(\min_i m_{d-1}(\partial V_i^\epsilon)\right)^3$$

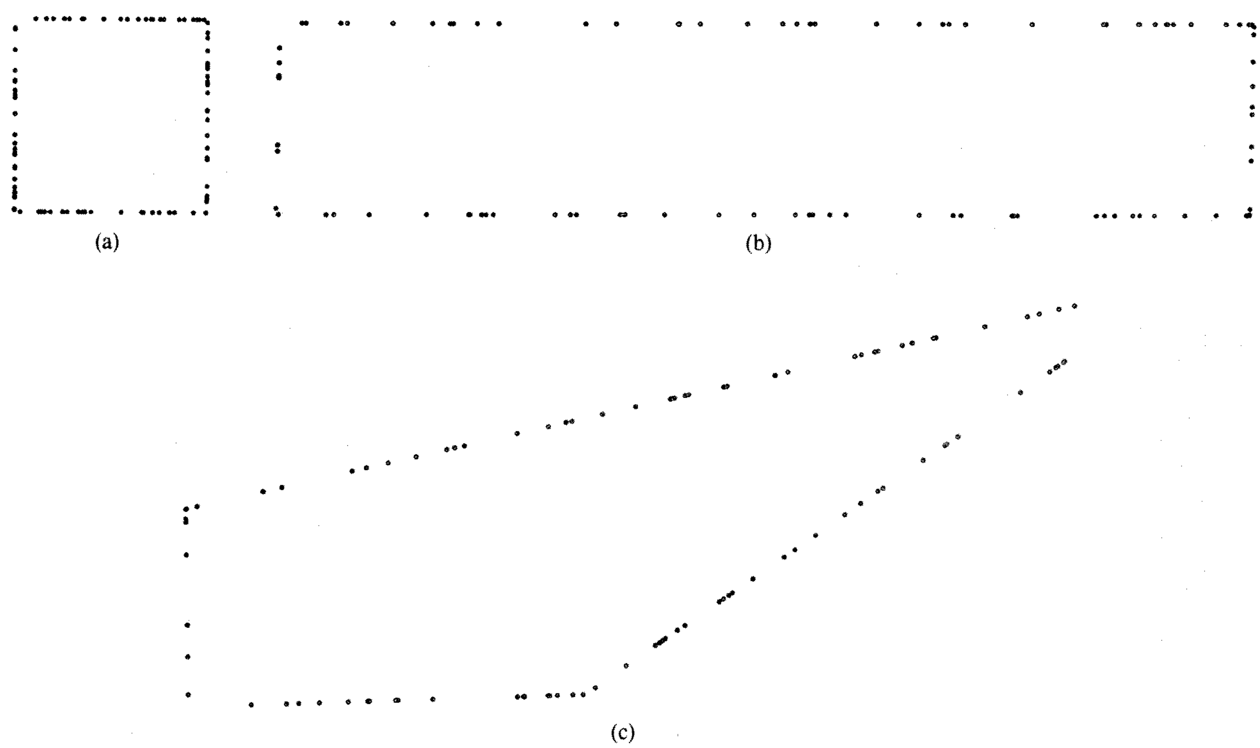


Figure 5. One hundred iteration points.

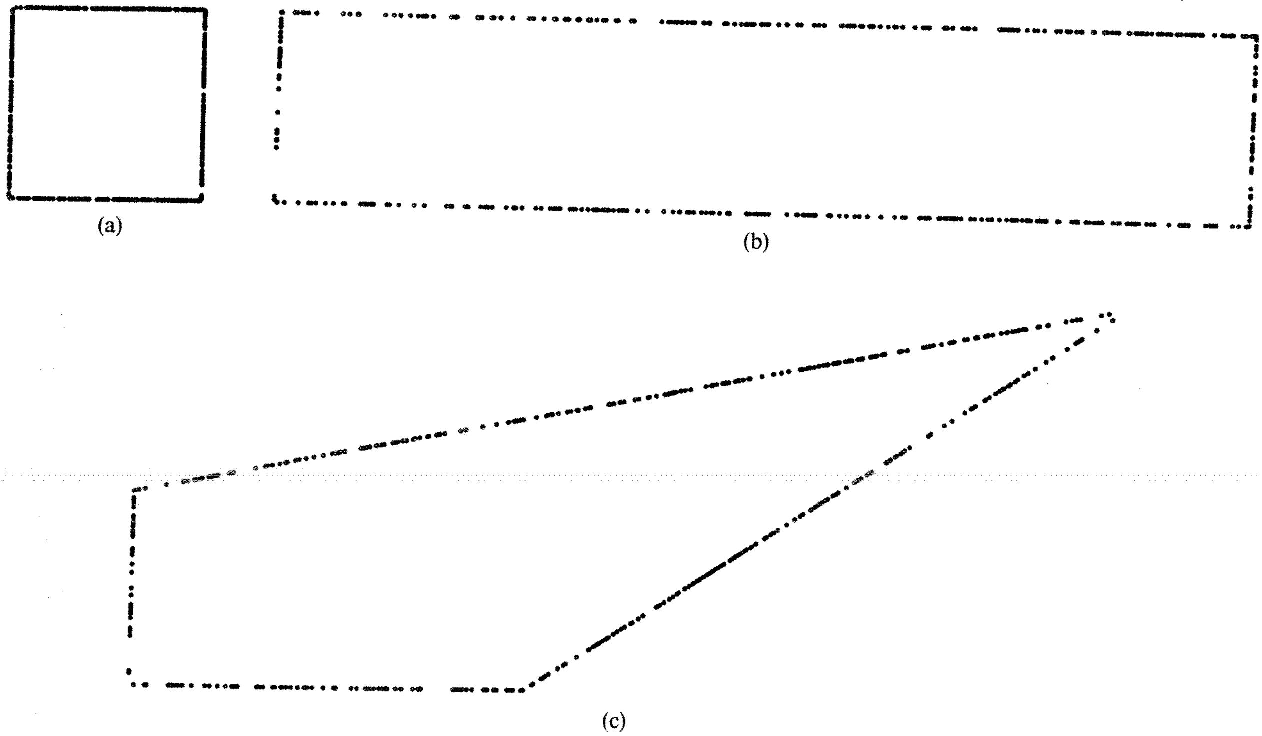


Figure 6. Five hundred iteration points.

only depends on the region S, and *not* on the particular algorithm used. This means that we can compare the convergence rates of the SB algorithms by comparing the values of δ_h. Recall from Section 1 that

$$\delta_{h_O} = \frac{1}{C_d}, \quad \delta_{h_L} = \frac{2}{C_d}, \quad \text{and } \delta_{h_R} = \frac{1}{B_{d-1}}.$$

It is easy to see that for $d \geq 2$, the following holds:

$$\delta_{h_O} < \delta_{h_L} < \delta_{h_R}.$$

Moreover, as the function $h_R(x, y)$ is a constant, and the move probability equals one for this algorithm, we have

$$\delta_h \leq \delta_{h_R}$$

for all SB algorithms, since $\int_{\partial S} p(y \mid x)\, dy \leq 1$ for all SB algorithms. Thus, taking into account that we are discussing lower bounds, we may conjecture that the sequence of iteration points generated by running SB converges faster to the uniform distribution than the sequence of points generated by limping SB, which, in turn, converges faster to the uniform distribution than the sequence generated by original SB. Furthermore, running SB has the fastest rate of convergence of all SB algorithms. It should, however, be noted that for a specific region S it might be possible to obtain a better lower bound than the one given in Theorem 3, specifically for algorithms having a nonconstant function h associated with them (like, for instance, original SB). Therefore, we should be careful in drawing strong conclusions from only the comparison of values for δ_h, when h is not equal to a constant.

3. EXPERIMENTAL RESULTS

We conclude the paper with some experimental results of the algorithm running SB. We let Figures 5 and 6 speak for themselves.

ACKNOWLEDGMENT

This paper is based on work supported by NATO Collaborative Research grant no. 0119/89. The work of the second author was supported by NSERC grant A8807, while the work of the third author was supported by NSERC grant A4625.

REFERENCES

Berbee, H. C. P., C. G. E. Boender, A. H. G. Rinnooy Kan, C. L. Scheffer, R. L. Smith and J. Telgen. 1987. Hit-and-Run Algorithms for the Identification of Nonredundant Linear Inequalities. *Math. Prog.* **37,** 184–207.

BOENDER, C. G. E., R. J. CARON, J. F. MCDONALD, R. L. SMITH AND J. TELGEN. 1988a. A Class of Shake-and-Bake Algorithms for Generating Random Points on the Surface of a Bounded Region. Technical Report WMR 88-08, Department of Mathematics and Statistics, University of Windsor, Ontario, Canada.

BOENDER, C. G. E., A. H. G. RINNOOY KAN, H. E. ROMEIJN AND A. C. F. VORST. 1988b. Shake-and-Bake Algorithms for Generating Uniform Points on the Boundary of Bounded Polyhedra. Technical Report 8826/A, Econometric Institute, Erasmus University, Rotterdam, The Netherlands.

DOOB, J. L. 1953. *Stochastic Processes*. John Wiley, New York.

KNUTH, D. E. 1969. *The Art of Computer Programming*, Vol. 2. Addison-Wesley, Reading, Mass.

PATEL, N. R., AND R. L. SMITH. 1983. The Asymptotic Extreme Value Distribution of the Sample Minimum of a Concave Function Under Linear Constraints. *Opns. Res.* **31**, 789–794.

SMITH, R. L. 1982. Private communication.

SMITH, R. L. 1984. Efficient Monte Carlo Procedures for Generating Random Feasible Points Uniformly Over Bounded Regions. *Opns. Res.* **32**, 1296–1308.

SMITH, R. L., AND J. TELGEN. 1981. Random Methods for Identifying Nonredundant Constraints. Technical Report 81-4, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor.