Variants of Transformed Density Rejection and Correlation Induction

Josef Leydold[1], Erich Janka[1*], and Wolfgang Hörmann[2**]

[1] Department for Statistics, WU Wien, Augasse 2–6, A-1090 Vienna, Austria
email: `Josef.Leydold@statistik.wu-wien.ac.at` and
`janka@statistik.wu-wien.ac.at`

[2] IE Department, Boğaziçi University Istanbul, 80815 Bebek-Istanbul, Turkey
email: `hormannw@boun.edu.tr`

Abstract. In this paper we present some variants of transformed density rejection (TDR) that provide more flexibility (including the possibility to halve the expected number of uniform random numbers) at the expense of slightly higher memory requirements. Using a synchronized first stream of uniform variates and a second auxiliary stream (as suggested by Schmeiser and Kachitvichyanukul (1990)) TDR is well suited for correlation induction. Thus high positive and negative correlation between two streams of random variates with same or different distributions can be induced.

1 Introduction

Transformed density rejection (TDR) has been introduced by Devroye (1986) and under a different name by Gilks and Wild (1992), and generalized by Hörmann (1995). It is an acceptance/rejection method that is based on the fact that many probability density functions f can be transformed by strictly monotonically increasing differentiable functions T such that $T \circ f$ is concave. Such densities (and their corresponding distributions) are called *T-concave*; *log-concave* distributions are examples with $T = \log$. Then it is easy to construct a majorizing function for the transformed density by the minimum of several tangents. *Squeezes* are obtained by secants between the touching points of these tangents. By transforming back into the original scale we get a *majorizing* or *hat* function and *squeezes* for the density f (see figure 1).

Of course $T\colon (0,\infty) \to \mathbb{R}$ must satisfy some properties (Hörmann 1995):

(1) $T \circ f$ is concave.
(2) T' exists and is greater than 0 for all x; thus T^{-1} exists.
(3) The area below the hat function must be finite.

Additionally

* Research supported by the Austrian Science Foundation (FWF), project no. P12805-MAT

** Research supported by the Austrian Academy of Science, APART-scholarship.

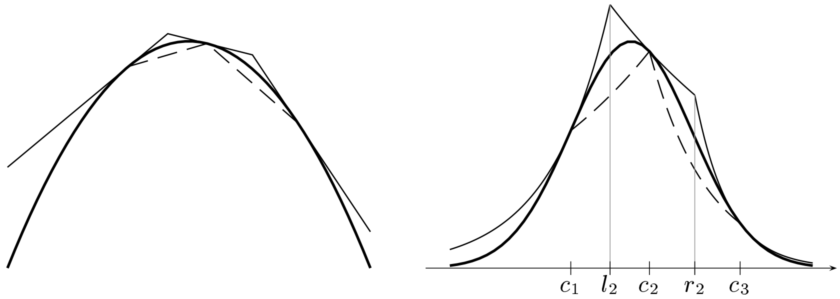

Fig. 1. Hat function and squeeze with three points of contact. Transformed scale (l.h.s.) and original scale (r.h.s.)

(4) It should be easy to generate from the hat distribution.

Hörmann (1995) suggests the following family of transformations T_c:

$$T_0(x) = \log(x), \quad \text{and} \quad T_c(x) = \operatorname{sign}(c)\, x^c, \text{ for } c \neq 0.$$

Parameters $c > 0$ and $c \leq -1$ are only possible if the domain of f is bounded. For computational reasons $c = -\frac{1}{2}$ (if possible) is suggested. It is important to note that a T_c-concave density is also $T_{c'}$-concave for every $c' \leq c$ and that the performance of the algorithm becomes worse for decreasing c (Hörmann 1995). For $c = -\frac{1}{2}$ the class of T_c-concave distributions contains most bounded unimodal densities with subquadratic tails and therefore practically all standard distributions.

We give a detailed description of the original method as suggested by Gilks and Wild (1992) in the next section. Although this algorithm is very flexible and fast there exist other universal methods that require less uniform random variates like a version of the ratio-of-uniforms method by Leydold (2000a) or a method with piecewise constant hat by Ahrens (1993, 1995). Leydold and Hörmann (2001a) give a short survey. In section 3 we describe variants of TDR. The modifications of the algorithm are subtle but have remarkable consequences on the performance and reduce some numerical problems. In section 4 we show that these extremely fast algorithms can be used for correlation induction applying the ideas of Schmeiser and Kachitvichyanukul (1990).

2 Transformed density rejection

Algorithm `tdrgw` describes the main steps of the basic method. There and in the following c_j denote the construction points of the tangents, s the squeeze and h the hat function. Let $I_j = [l_j, r_j]$ be the intervals where h

is given by the tangent at c_j. The restrictions of the hat h to the interval I_j are denoted by $h_j = h|_{I_j}$; analogously $s_j = s|_{I_j}$. The "c.d.f." of h_j is denoted by $H_j(x) = \int_{l_j}^{x} h_j(t)\,dt$. The area below the hat h in an interval is given by $A_j = \int_{I_j} h(t)\,dt$. Notice that $(T \circ h_j)(x) = \alpha_j + \beta_j\,(x - c_j)$ is a linear function in I_j, where $\alpha_j = (T \circ f)(c_j)$ and $\beta_j = (T \circ f)'(c_j)$. Thus $h_j(x) = T^{-1}(\alpha_j + \beta_j\,(x - c_j))$. The squeeze $T \circ s_j$ splits into two linear functions on either side of the construction point c_j which can be computed by means of α_{j-1}, α_j and α_{j+1}.

Algorithm `tdrgw`

Require: density $f(x)$; transformation $T(x)$, construction points $c_1, \dots, c_n$.

/* Setup: Construct hat $h(x)$ and squeeze $s(x)$ */
1: Compute $\alpha_j = (T \circ f)(c_j)$ and $\beta_j = (T \circ f)'(c_j)$ for each $j = 1, \dots, n$.
2: Compute intervals $I_j = [l_j, r_j]$ for each j by intersecting the tangents.
3: Compute areas $A_j = \int_{l_j}^{r_j} h_j(t)\,dt$ for each j.
4: $A \leftarrow A_1 + \cdots + A_n$.
/* Generator */
5: **loop**
6: Generate $V \sim U(0, A)$.
/* Generate J with probability vector proportional to $(A_1, \dots, A_n)$. */
7: $J \leftarrow \min\{j : A_1 + \cdots + A_j \geq V\}$.
/* Generate X with density proportional to h_J. */
8: $V \leftarrow V - (A_1 + \cdots + A_J)$.
9: $X \leftarrow H_J^{-1}(V)$.
/* Accept or reject */
10: Generate $U \sim U(0, 1)$.
11: **if** $U\,h(X) \leq s(X)$ **then** /* evaluate squeeze */
12: **return** X.
13: **if** $U\,h(X) \leq f(X)$ **then** /* evaluate density */
14: **return** X.

Step 7 is executed in constant time by means of *indexed search* (Chen and Asau 1974). Notice that the random variate X is generated by inversion, when the algorithm is implemented properly. This will be important for correlation induction. Therefore the usage of the alias method by Walker (1974) is not recommended. In our experiments there was practically no difference in speed anyway.

`tdrgw` requires two uniform random numbers per iteration. The performance of algorithm `tdrgw` can be estimated (and controlled) by the ratio

$$\varrho = \frac{\text{area below hat}}{\text{area below squeeze}}.$$

It is an upper bound for the rejection constant. The number $(\varrho - 1)$ gives approximately the expected number of evaluations of the density function per generated random variate. For ϱ close to 1 the algorithm is fast and the marginal generation time does not depend on the given density f. Moreover it is close to inversion from the c.d.f. and thus inherits many of the desired properties of the inversion method (but is much faster for almost all distributions).

For the problem of finding appropriate construction points c_j, Gilks and Wild (1992) have introduced the concept of *adaptive rejection sampling*: (1) Start with at least two points of contact, one on either side of the mode and start to generate variates. (2) Whenever the density function has to be evaluated at a point x, then x is added as new construction point and the hat is updated, until a certain stopping criterion (an upper bound for ratio ϱ or the maximal number of construction points) is satisfied. For finding optimal construction points Derflinger, Hörmann, and Tirler (2001) suggest a very efficient asymptotic method. Leydold (2000a) has reported good performance with the rule of thumb of "equidistributed angles":

$$c_i = \tan(-\pi/2 + i\,\pi/(n+1)), \qquad i = 1, \dots, n.$$

These are at least good starting points for adaptive rejection sampling.

TDR is a very flexible method and that works for a large class of distributions. Evans and Swartz (1998) have even suggested an adaption that works for distributions that are not T-concave provided that the inflection points of the transformed density are known.

3 Variants

3.1 Proportional squeezes

Evans and Swartz (1998) have used secants between the boundary points of the intervals I_j. This reduces some programming complexity because $T \circ s_j$ is a linear function in I_j. However it increases ϱ a little bit for given construction points c_j. Moreover additional evaluations of the density function f (at l_j and r_j) are necessary in the set-up. We can simplify this approach at the price of a slightly increased ϱ by using a squeeze that is proportional to the hat h_j in the interval I_j (see figure 2). In detail we set $s_j(x) = \nu_j\, h_j(x)$ for a proper constant ν_j. The consequences are that we have a faster rejection step. No squeeze has to be computed, only U has to be compared to the constant ν_j of proportionality. This can even be done *before* X is generated in I_j.

Obviously the best choice for ν_j is given by $\nu_j = \min_{x \in I_j} f(x)/h_j(x)$. When using the transformation T_c, it is easy to find ν_j. In the transformed scale the squeeze $T_c(s_j(x)) = T_c(\nu_j\, h_j(x))$ is again a linear function. Thus by the concavity of the transformed density we only have to evaluate $f(x)/h(x)$ at the boundary points of the interval I_j. The value of ν_j is then the least

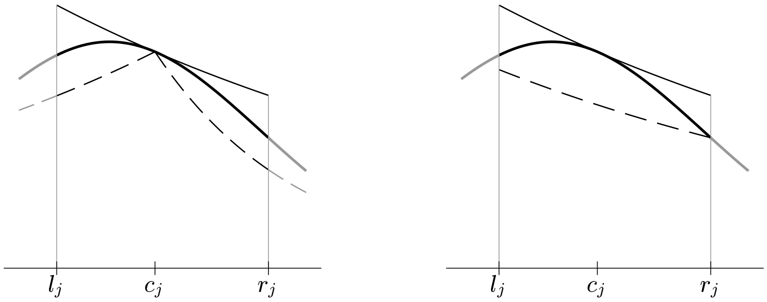

Fig. 2. Hat function and squeeze as used in the original (l.h.s.) and the new (r.h.s.) algorithm.

of these two ratios (of course $\nu_j = 0$ if I_j is not bounded). Algorithm `tdrps` uses this new squeeze. Notice that, at the expense of step 3, we neither have to evaluate the hat nor the squeeze in step 12.

Algorithm `tdrps`

Require: density $f(x)$; transformation $T(x)$, construction points $c_1, \ldots, c_n$.

/* Setup: Construct hat $h(x)$ and squeeze $s(x)$ */
1: Compute $\alpha_j = (T \circ f)(c_j)$ and $\beta_j = (T \circ f)'(c_j)$ for each $j = 1, \ldots, n$.
2: Compute intervals $I_j = [l_j, r_j]$ for each j.
3: Compute $\nu_j = \min_{x \in \{l_j, r_j\}} f(x)/h_j(x)$ for each j.
4: Compute areas $A_j = \int_{I_j} h_j(t)\, dt$ for each j.
5: $A \leftarrow A_1 + \cdots + A_n$.
/* Generator */
6: **loop**
7: Generate $V \sim U(0, A)$.
8: $J \leftarrow \min\{j \colon A_1 + \cdots + A_j \geq V\}$.
9: $V \leftarrow V - (A_1 + \cdots + A_J)$.
10: $X \leftarrow H_J^{-1}(V)$.
11: Generate $U \sim U(0, 1)$.
12: **if** $U \leq \nu_J$ **then** /* evaluate squeeze */
13: **return** X.
14: **if** $U\, h(X) \leq f(X)$ **then** /* evaluate hat and density */
15: **return** X.

Due to the different construction of the squeeze we have to reconsider the choice of the construction points. The rule of thumb in eq. (2) still works. But we have to change the rule for adaptive rejection sampling. Now we add a new construction point x whenever a point x is *rejected.* (It would not make sense to add a point whenever we have to evaluate the density function as in the original algorithm.)

It is an open problem whether a method for finding optimal construction points similar to Derflinger et al. (2001) exists. Nevertheless the construction points generated by this algorithm are very good points, especially when ϱ is close to 1.

This variant of TDR can also be adapted for distributions that are not T-concave provided that the inflection points of the transformed density are known by using the idea of Evans and Swartz (1998). As a first step we have to split the domain of the density at these inflection points. In intervals where the transformed density is convex the tangent is used to construct the squeeze $s_j(x)$ and the hat is given by $h_j(x) = \nu_j h_j(x)$ where $\nu_j \geq 1$ choosen analogously to the concave intervals as described above.

3.2 Region of immediate acceptance

When we use a squeeze that is proportional to the hat h_j as suggested above, the region between hat and squeeze is much simpler than in the original algorithm `tdrgw`. Now random variates with density proportional to the difference between hat and squeeze $(h - s)$ can easily be generated, since for an interval I_j we find $h(x) - s(x) = h(x) - \nu_j\, h(x) = (1 - \nu_j)\, h(x)$, i.e. its density is proportional to the hat $h(x)$ (in contrast to the original version, see Leydold (2000b)). Thus we can decompose the hat h into the density below the squeeze and the density between squeeze and hat, where both densities are proportional to the hat function in every interval I_j. The area below the squeeze can be seen as "region of immediate acceptance", i.e. the generated random variate X can be immediately returned without the requirement of a second uniform random number. Therefore we can compile an algorithm where the expected number of uniform random numbers is close to one when ϱ is close to one.

Algorithm `tdria`

Require: density $f(x)$; transformation $T(x)$, construction points $c_1, \ldots, c_n$.

```
/* Setup: Construct hat h(x) and squeeze s(x) */
Compute α_j = (T ∘ f)(c_j) and β_j = (T ∘ f)'(c_j) for each j = 1,...,n.
Compute intervals I_j = [l_j, r_j] for each j.
Compute ν_j = min_{x∈{l_j,r_j}} f(x)/h_j(x) for each j.
Compute areas A_j = ∫_{I_j} h_j(t) dt for each j.
A ← A_1 + ··· + A_n.
/* Generator */
loop
    Generate V ~ U(0, A).
    J ← min{j: A_1 + ··· + A_j ≥ V}.
    V ← V − (A_1 + ··· + A_J).
```

if $V \leq \nu_J A_J$ **then** /* below squeeze, immediate acceptance */
  $X \leftarrow H_J^{-1}(V/\nu_J)$.
  return X.
else
  $X \leftarrow H_J^{-1}((A-V)/(1-\nu_J))$.
  Generate $U \sim U(0,1)$.
  if $U\,h(X) \leq f(X)$ **then** /* evaluate density */
    return X.

4 Correlation induction

Common random numbers and antithetic variates are two of the best known variance reduction methods for simulation experiments (see e.g. Bratley, Fox, and Schrage (1983)). Using the inversion method it is no problem to induce the strongest possible positive or negative correlation when generating two random variate streams (even with different distributions). For positive correlation (*common random numbers*) we simply use the same uniform numbers for both streams, for negative correlation (*antithetic variates*) we take U_i for the generation of the first stream and $1-U_i$ for the second stream. This is one of the main reasons that Bratley, Fox, and Schrage (1983) call inversion the method of choice for generating random variates. On the other hand inversion often is not possible algebraically and numerical inversion is extremely slow, for example about hundred times slower than a good rejection algorithm for the Beta distributions.

Therefore Schmeiser and Kachitvichyanukul (1990) have suggested rules to realize correlation induction for rejection algorithms. The results of that paper and of Hörmann and Derflinger (1994) show that this idea only works well if the rejection constant is close to one. Using TDR we have no problems to get such rejection algorithms. Thus algorithms based on TDR should be well suited for correlation induction.

Formulating the rules of Schmeiser and Kachitvichyanukul (1990) for our situation we get:

(1) Conceptually, two independent uniform random number streams are used. If not needed the second stream is ignored.
(2) For each random variate generated, the same number n_1 of random numbers from the first stream are used.
(3) The first of these random numbers is used to sample from the hat distribution by inversion.
(4) The second stream is used if all of the random numbers from the first stream are exhausted by rejection. Thus the number of random numbers from the second stream is random, the expected number should be small.

Rule (2) is necessary to guarantee that the streams of uniform random numbers for both random variables always run synchronously, whereas rule (3) tries to induce correlation between corresponding random variables, when the first number is accepted for both generators. (This only works well if inversion is used to sample from the hat-function in the rejection algorithms.)

The pair of generated random variates is uncorrelated only when rejection occurs for at least one of the two variate generators. However this is extremely rare and its frequency is less than $2(\varrho - 1)$.

The choice of n_1 (the fixed number of uniforms from the first stream used for generating one variate) depends mainly on the rejection algorithms used. Using $n_1 = 4$ (as suggested by Schmeiser and Kachitvichyanukul (1990)) is a waste of uniform random numbers for ϱ close to one. We mainly experimented with n_1 such that n_1 is the number of uniform random numbers required for one acceptance/rejection loop in the corresponding algorithm, i.e., we set $n_1 = 2$ for `tdrps` and $n_1 = 1$ for `tdria`.

5 Computational experience

We expected from the simulation results reported by Schmeiser and Kachitvichyanukul (1990) and Hörmann and Derflinger (1994) that TDR works well for correlation induction. So we hoped before starting this work that the results of our experiments will underline this nice property and will also help to solve the question whether variant `PS` or `IA` works better for correlation induction. Notice that due to immediate acceptance there is a conceptual difference between `PS` (TDR with proportional squeeze) which is a pure rejection algorithm and `IA` (TDR with immediate acceptance) an algorithm combining decomposition and rejection. Therefore using `PS`, points are generated by inversion from an approximate distribution (hat) and few points are canceled out (by rejection) to correct the distribution. For `IA` (small) holes are cut out of the interval $(0, 1)$ to get a smaller interval which is used to generate from an approximate distribution (squeeze). In the case that immediate acceptance is not possible, few additional points are added to obtain the correct distribution. Numerical inversion (`NI`) is the method that is known to perform optimal for correlation induction but it is very slow.

Table 1 contains the distributions we used for our experiments and gives the marginal execution times when the transformation $T(x) = -1/\sqrt{x}$ is used. For $T(x) = \log(x)$ the marginal generation times are about two times slower for `PS` and `IA`, and about three times slower for `GW`. It clearly shows that numerical inversion is so slow that it should not be used for large simulations. The results `GW` (original version by Gilks and Wild (1992) as described in algorithm `tdrgw`) are included to demonstrate that the two new variants are clearly faster. Compare with 0.38 µs for exponential distribution by inversion and 0.46 µs for the Box-Muller method for normal distribution to see the very good speed of TDR. Note that the speed is almost the same for all

Distribution			(N)	GW	PS	IA	NI
N	...	standard normal	(29)	0.52	0.38	0.36	4.2
E	...	exponential	(14)	0.53	0.38	0.35	4.2
Γ	...	gamma with s.p. $\alpha = 2$	(26)	0.53	0.39	0.37	12.5
B_1	...	beta with s.p. 1 and 2	(12)	0.54	0.40	0.38	11.0
B_2	...	beta with s.p. 10 and 20	(29)	0.54	0.41	0.38	60.9

Table 1. Timings (in µs). Sample size $= 10^5$.
`GW`, `PS`, `IA`: necessary number N of construction points to reach $\varrho = 1.01$ with $c = -\frac{1}{2}$. Same (optimal) construction points are used for all three methods.
`NI`: auxilliary table of size 100, accuracy $\epsilon = 10^{-8}$.
For comparison: 0.46 µs for Box-Muller method, 0.38 µs for exponential distribution via logarithm.

tested distributions; but `GW` is more than 30% slower than `PS` or `IA`. The speed difference between `PS` and `IA` comes mainly from the lower number of uniform deviates required by `IA` (and is greater on other computers). Decreasing ϱ to values smaller than 1.01 by increasing the number N of construction points has practically no influence on the speed for the given distribution. Notice that the table compares black box algorithms. It is obvious that special routines for computing the inverse of the c.d.f. of some of these distributions exist that are much faster. On the other hand we have used tables for numerical inversion that speed up the generation at the expense of a setup that is much slower than that for the TDR variants.

Tables 2 and 3 give the results of our correlation induction experiments. Again we have used the transformation $T(x) = -1/\sqrt{x}$ for the TDR algorithms (`PS` with $n_1 = 2$ and `IA` with $n_1 = 1$) and compared the results with the correlation obtained by numerical inversion (`NI`). The construction points for the hat functions are computed by the algorithm by Derflinger, Hörmann, and Tirler (2001). (Thus the real value of ϱ differs a little bit from 1.11 and 1.01 respectively.) Results for $T(x) = \log(x)$ are slightly better, since the tails of the hat distributions are not so heavy then. Moreover less construction points are required for the same ϱ. We have used the Mersenne Twister by Matsumoto and Nishimura (1998) with the same seed as source of the first stream of uniform random numbers and `tt800` by Matsumoto and Kurita (1994) for the second auxilliary stream (both are implemented in the *PRNG* library by Lendl (2001)). We have generated 10^5 pairs with strong positive correlation (common random numbers) and strong negative correlation (antithetic variates). We always used the same generation method for both streams.

Notice that the second auxilliary stream of uniform random numbers is not synchronized. Thus the correlation between streams of random variates of the same distribution is not 1. It would not be a problem to syncronize the second stream, too. However the chosen setting of our experiments shows

Correlation: $+$, $\rho = 1.01$

		N	E	Γ	B_1	B_2	U
N	PS	0.99					
	IA	0.99					
	NI	1					
E	PS	0.89	0.99				
	IA	0.89	0.98				
	NI	0.90	1				
Γ	PS	0.94	0.98	0.99			
	IA	0.93	0.97	0.98			
	NI	0.94	0.99	1			
B_1	PS	0.96	0.93	0.96	0.99		
	IA	0.96	0.92	0.96	0.99		
	NI	0.97	0.94	0.97	1		
B_2	PS	0.99	0.91	0.95	0.97	0.99	
	IA	0.98	0.91	0.95	0.97	0.99	
	NI	0.99	0.92	0.96	0.98	1	
U	PS	0.97	0.85	0.91	0.97	0.97	1
	IA	0.97	0.86	0.91	0.97	0.97	1
	NI	0.97	0.86	0.91	0.97	0.97	1

Correlation: $+$, $\rho = 1.11$

		N	E	Γ	B_1	B_2	U
N	PS	0.96					
	IA	0.91					
	NI	1					
E	PS	0.80	0.95				
	IA	0.77	0.85				
	NI	0.90	1				
Γ	PS	0.88	0.84	0.96			
	IA	0.84	0.83	0.89			
	NI	0.94	0.99	1			
B_1	PS	0.90	0.87	0.90	0.97		
	IA	0.87	0.84	0.87	0.94		
	NI	0.97	0.94	0.97	1		
B_2	PS	0.94	0.82	0.89	0.91	0.96	
	IA	0.89	0.80	0.86	0.89	0.91	
	NI	0.99	0.92	0.96	0.98	1	
U	PS	0.92	0.82	0.86	0.94	0.92	1
	IA	0.91	0.80	0.87	0.93	0.92	1
	NI	0.97	0.86	0.91	0.97	0.97	1

Table 2. Observed induced correlation for common random numbers (see Tab. 1, U denotes uniform distribution).

Correlation: $-$, $\rho = 1.01$

		N	E	Γ	B_1	B_2	U
N	PS	-0.98					
	IA	-0.98					
	NI	-1					
E	PS	-0.88	-0.63				
	IA	-0.89	-0.63				
	NI	-0.90	-0.64				
Γ	PS	-0.93	-0.71	-0.78			
	IA	-0.93	-0.71	-0.78			
	NI	-0.94	-0.72	-0.79			
B_1	PS	-0.96	-0.78	-0.85	-0.92		
	IA	-0.96	-0.78	-0.85	-0.92		
	NI	-0.97	-0.79	-0.86	-0.93		
B_2	PS	-0.98	-0.86	-0.91	-0.95	-0.98	
	IA	-0.98	-0.86	-0.91	-0.95	-0.98	
	NI	-0.99	-0.87	-0.92	-0.96	-0.99	
U	PS	-0.97	-0.86	-0.91	-0.97	-0.97	-1
	IA	-0.97	-0.86	-0.91	-0.97	-0.97	-1
	NI	-0.97	-0.86	-0.91	-0.97	-0.97	-1

Correlation: $-$, $\rho = 1.11$

		N	E	Γ	B_1	B_2	U
N	PS	-0.87					
	IA	-0.88					
	NI	-1					
E	PS	-0.77	-0.60				
	IA	-0.76	-0.55				
	NI	-0.90	-0.64				
Γ	PS	-0.82	-0.65	-0.70			
	IA	-0.82	-0.63	-0.70			
	NI	-0.94	-0.72	-0.79			
B_1	PS	-0.88	-0.73	-0.79	-0.88		
	IA	-0.86	-0.71	-0.77	-0.86		
	NI	-0.97	-0.79	-0.86	-0.93		
B_2	PS	-0.87	-0.76	-0.80	-0.87	-0.87	
	IA	-0.88	-0.73	-0.81	-0.86	-0.88	
	NI	-0.99	-0.87	-0.92	-0.96	-0.99	
U	PS	-0.91	-0.81	-0.86	-0.95	-0.92	-1
	IA	-0.91	-0.80	-0.86	-0.94	-0.92	-1
	NI	-0.97	-0.86	-0.91	-0.97	-0.97	-1

Table 3. Observed induced correlation for antithetic random variates

the (small) deviation from the optimal correlation caused by the rejection algorithm.

The results in Tables 2 and 3 show that for $\varrho = 1.01$ the induced correlation is practically the same for all three methods as the difference is never larger than 0.02.

The case $\varrho = 1.11$ was included to see if there are the differences between `PS` and `IA`. For positive correlation `PS` is slightly better than `IA`, for negative correlation there is hardly a difference. This shows that the choice of the variant of TDR has practically no influence on correlation induction. There is one difference between $\varrho = 1.01$ and $\varrho = 1.11$. For the first about 15–30 construction points are necessary (as we used optimal construction points) and the marginal generation times is almost independent from the distribution (see Tab. 1). When $\varrho = 1.11$ only 4–8 construction points are required but the marginal generation times are slower and depend on the distribution.

6 Conclusion

We have demonstrated in this paper that the two new variants of transformed density rejection are well suited to generate correlated random variates. For many construction points (ϱ close to 1) TDR produces sequences with practically the same correlation as numerical inversion. If ϱ is not too close to 1 then pure rejection induces higher correlation than the composition method.

The difference in speed between the TDR algorithms and numerical inversion is so large that for some common distributions the total time of a simulation can be more than doubled even if the random variate generation (with rejection algorithms) takes only one percent of the total execution time.

We have coded these (and many other) algorithms and compiled in a library called UNIVERSAL NONUNIFORM RANDOM variate generators (UNURAN) (Leydold and Hörmann 2001b). It is entirely written in ANSI C using an object oriented programming interface. Also correlation induction can easily be handled using this interface.

References

Ahrens, J. H. (1993). Sampling from general distributions by suboptimal division of domains. *Grazer Math. Berichte 319*, 20 pp.

Ahrens, J. H. (1995). A one-table method for sampling from continuous and discrete distributions. *Computing 54*(2), 127–146.

Bratley, P., B. L. Fox, and E. L. Schrage (1983). *A Guide to Simulation.* New York: Springer-Verlag.

Chen, H. C. and Y. Asau (1974). On generating random variates from an empirical distribution. *AIIE Trans. 6*, 163–166.

Derflinger, G., W. Hörmann, and G. Tirler (2001). The optimal selection of hat functions for rejection algorithms. in preparation, private communication.

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. New-York: Springer-Verlag.

Evans, M. and T. Swartz (1998). Random variable generation using concavity properties of transformed densities. *Journal of Computational and Graphical Statistics 7*(4), 514–528.

Gilks, W. R. and P. Wild (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics 41*(2), 337–348.

Hörmann, W. (1995). A rejection technique for sampling from T-concave distributions. *ACM Trans. Math. Software 21*(2), 182–193.

Hörmann, W. and G. Derflinger (1994). Universal generators for correlation induction. In R. Dutter and W. Grossmann (Eds.), *Compstat, Proceedings in Computational Statistics*, Heidelberg, pp. 52–57. Physica-Verlag.

Lendl, O. (2001). *prng 3.0 – A library for the generation of pseudorandom numbers*. Hellbrunnerstr. 34, A-5020 Salzburg, Austria: Institute of Mathematics, Paris-Lodron University Salzburg. available at `http://random.mat.sbg.ac.at/ftp/pub/software/gen/`.

Leydold, J. (2000a). Automatic sampling with the ratio-of-uniforms method. *ACM Trans. Math. Software 26*(1), 78–98.

Leydold, J. (2000b). A note on transformed density rejection. *Computing 65*(2), 187–192.

Leydold, J. and W. Hörmann (2001a). Universal algorithms as an alternative for generating non-uniform continuous random variates. In G. I. Schuëller and P. D. Spanos (Eds.), *Monte Carlo Simulation*, pp. 177–183. A. A. Balkema. Proceedings of the International Conference on Monte Carlo Simulation 2000.

Leydold, J. and W. Hörmann (2001b). *UNURAN – A library for Universal Non-Uniform Random number generators*. A-1090 Wien, Austria: Institut für Statistik, WU Wien. available at `http://statistik.wu-wien.ac.at/unuran/`.

Matsumoto, M. and Y. Kurita (1994). Twisted GFSR generators II. *ACM TOMACS 4*(3), 254–266.

Matsumoto, M. and T. Nishimura (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM TOMACS 8*(1), 3–30.

Schmeiser, B. W. and V. Kachitvichyanukul (1990). Non-inverse correlation induction: guidelines for algorithm development. *J. Comput. Appl. Math. 31*, 173–180.

Walker, A. J. (1974). New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electron. Lett. 10*, 127–128.