行政院國家科學委員會專題研究計畫 成果報告

貝他分配演算法之評估
研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 97-2118-M-004-004-
執 行 期 間 ：97 年 08 月 01 日至 98 年 09 月 30 日
執 行 單 位 ：國立政治大學統計學系

計 畫 主 持 人 ：洪英超

計畫參與人員：碩士班研究生-兼任助理人員：鄭經維
碩士班研究生-兼任助理人員：李佳燁
碩士班研究生-兼任助理人員：賴昱瑋

報 告 附 件 ：出席國際會議研究心得報告及發表論文

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 98 年 12 月 25 日

報告內容

(一) 前言及文獻探討

Beta variates are used extensively in Bayesian statistics, stochastic modeling and simulation, program evaluation and review techniques (PERT), critical path method (CPM), and project management and control systems (for more applications, see Gupta and Nadarajah, 2004; Morgan and Henrion, 1990). A standard beta variate $Beta(\alpha, \beta)$ has the probability density function

$$f(x \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \le x \le 1, \tag{1}$$

with $\alpha > 0$ and $\beta > 0$ being shape parameters. Over the last two decades, the statistical/mathematical software packages (such as R, Splus, SAS, SPSS, and Matlab) have been well developed for generating various types of random numbers.

These software packages are useful for fundamental simulation studies, but prove inefficient for research involving complex computer codes and requiring a large amount of random number generation.

Over the years, many algorithms have been introduced in the literature for the computer generation of beta variates. Among them, a large class of algorithms are based on acceptance/rejection methods. The well-known ones are: Jöhnk's method (Jöhnk, 1964) and Forsythe's method (Forsythe, 1972), which are valid for all $\alpha, \beta > 0$; the BN algorithm by Ahrens and Dieter (1974), which is valid for $\alpha, \beta > 1$; two switching algorithms that are valid for $\min(\alpha, \beta) < 1$, and a general switching algorithm that is valid for all $\alpha, \beta > 0$ (Atkinson, 1979; Atkinson and Whittaker, 1976). Some of these methods were also studied by Atkinson and Pearce (1976), wherein a comparison of computer generation times was made. Cheng (1978) proposed different modified versions of Forsythe's method that are valid for $\alpha, \beta > 0$ (algorithm BA), $\alpha, \beta > 1$ (algorithm BB), and $\min(\alpha, \beta) \le 1$ (algorithm BC), respectively. Schmeiser and Shalaby (1980) developed three algorithms that are valid for $\alpha, \beta > 1$. These three algorithms are called BNM, B2P, and B4P, where BNM is a modification of the BN algorithm. A detailed comparison of the above beta algorithms was also carried out by Schmeiser and Shalaby (1980), wherein the BB algorithm was shown to be the fastest for heavily skewed distributions while the BNM algorithm was shown to be the fastest for heavy-tailed symmetric distributions. For more detailed discussions and comparisons, the readers can refer to work done by Cheng (1978), Devroye (1986), and Johnson et al. (1995).

Schmeiser and Babu (1980, 1983) proposed two algorithms that are valid for $\alpha, \beta > 1$. These two algorithms are called B2PE and B4PE, which are extensions of algorithm B2P and B4P, respectively. Sakasegawa (1983) proposed three algorithms that are valid for $\alpha, \beta < 1$ (algorithm B00), $\alpha < 1 < \beta$ or $\beta < 1 < \alpha$ (algorithm B01), and $\alpha, \beta > 1$ (algorithm B11). These algorithms all belong to the class of stratified rejection methods, in which piece-wise envlopes and exponentials are applied in the center and tails of the desired beta distribution, respectively. Zechner and Stadlober (1993) proposed two algorithms that are valid for $\alpha, \beta > 1$. These two algorithms are called BPRS and BPRB, which improve acceptance/rejection in the center of the desired beta distribution by using the idea of patchwork rejection. We now summarize some important results obtained in these studies. When $\alpha, \beta < 1$, algorithm B00 has the smallest computer generation time among all existing algorithms. When $\alpha < 1 < \beta$ or $\beta < 1 < \alpha$, algorithm B01 has the smallest computer generation time among all existing algorithms. When $\alpha, \beta > 1$, if one parameter is close to 1 and the other is large, algorithm B4PE has the smallest computer generation time; otherwise algorithm BPRS has the smallest computer generation time.

(二) 研究目的

Kennedy (1988) proposed a stochastic search procedure that asymptotically generates beta variates. However, one finds the following problems in implementing Kennedy's algorithm: (i) not all beta variates with $\alpha > 0$, $\beta > 0$ can be generated; and (ii) different parameter settings in the algorithm may (asymptotically) generate the same beta variate. Our goal here is to first identify the beta variates that can possibly be generated by Kennedy's algorithm. Next, for any valid α and β, we introduce an optimal parameter setting so that this algorithm can achieve the fastest speed of generation. In the last part, we evaluate the optimized version of Kenney's algorithm (called Kennedy's MK algorithm) by comparing with other beta algorithms and the default beta random number generators of three statistical/mathematical software packages (R, SAS, and Matlab), in terms of the following performance measures: (i) validity of choice of shape parameters (i.e., all (α, β) that can possibly be generated); (ii) computer generation time; (iii) initial set-up time; (iv) goodness of fit; and (v) amount of random number generation required. From these comparisons based on an empirical study, we present three guidelines for choosing the best suited beta algorithm.

(三) 研究方法

We first study Kennedy's algorithm for generating beta variates. Consider a stochastic search procedure starting with the interval $[A_0, B_0] = [0, 1]$. Suppose that at the nth stage, the search is confined to a random subinterval $[A_n, B_n]$. Let C_n and D_n be the minimum and maximum of k independent random points that are uniformly distributed on the subinterval $[A_n, B_n]$. The next subinterval $[A_{n+1}, B_{n+1}]$ is then taken to be $[C_n, B_n]$, $[A_n, D_n]$, or $[C_n, D_n]$ with probabilities p, q, and r, respectively, such that $p + q + r = 1$. It was shown by Kennedy (1988) and Johnson and Kotz (1990) that the limiting distribution of $[A_n, B_n]$ (as $n \to \infty$) is a beta distribution with shape parameters $\alpha = k(p + r)$ and $\beta = k(q + r)$.

Validity of Choice of Shape Parameters

Case 1. $\alpha < 1, \beta < 1$

From (2), it is clear that k does not exist when $\alpha + \beta < 1$, and $k = 1$ otherwise.

Case 2. $\alpha < 1, \beta \geq 1$

In this case, Eq. (2) reduces to $\lceil \beta \rceil \leq k \leq \lfloor \alpha + \beta \rfloor$. Therefore, k does not exist when $\lceil \beta \rceil > \lfloor \alpha + \beta \rfloor$, $k = \beta$ when $\beta \in \mathbb{Z}^+$, and $k = \lfloor \beta \rfloor + 1$ otherwise.

Case 3. $\alpha \geq 1, \beta < 1$

Since this is a symmetric version of Case 2, it follows that k does not exist when $\lceil \alpha \rceil > \lfloor \alpha + \beta \rfloor$, $k = \alpha$ when $\alpha \in \mathbb{Z}^+$, and $k = \lfloor \alpha \rfloor + 1$ otherwise.

Case 4. $\alpha \geq 1, \beta \geq 1$

There exists at least one solution with $k = \lceil \max(\alpha, \beta) \rceil$.

Figure 1 displays the result of Θ (up to the choice of $\alpha = 5, \beta = 5$) by summarizing the results in Cases 1–4 above.

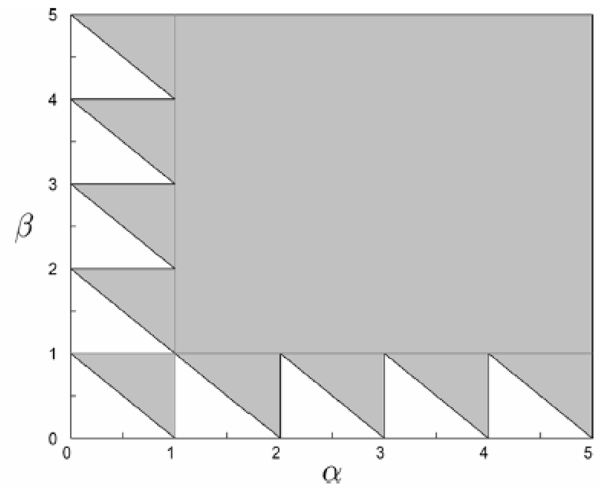

Figure 1. All possible values of (α, β) (i.e., the set Θ) that can be generated through Kennedy's algorithm (the grey part).

Note that from (2), one may have more than one choice of k for generating a particular beta variate with $(\alpha, \beta) \in \Theta$. So, the natural question now is how to choose the value of k so that the algorithm has the best performance in some sense.

(四) 結果與討論

Choosing Optimal k

For any given $(\alpha, \beta) \in \Theta$, the collection of all possible choices of k is given by

$$K(\alpha, \beta) = \left\{ k \in \mathbb{Z}^+ : \max\{\lceil \max(\alpha, \beta) \rceil, 1\} \leq k \leq \lfloor \alpha + \beta \rfloor \right\}. \tag{3}$$

In order to distinguish the search results of Kennedy's algorithm induced by different choices of k, let us denote the interval length at the nth stage by $L_n(k) = B_n(k) - A_n(k), k \in K(\alpha, \beta)$. Next, let us introduce a *tolerance level* Δ $(0 < \Delta < 1)$ so that the search stops as soon as $L_n(k) \leq \Delta$. This then introduces a stopping time (stage) of the algorithm as

$$N(k) = \min\{n \in \mathbb{Z}^+ : L_n(k) \leq \Delta\}, \quad k \in K(\alpha, \beta). \tag{4}$$

The following theorem explains how the value of k can be chosen so that the expected stopping time of Kennedy's algorithm is minimized.

Theorem 2.1. *For any given* $(\alpha, \beta) \in \Theta$, $0 < \Delta < 1$, *and* $k_0 = \min\{k : k \in K(\alpha, \beta)\}$,

$$E[N(k)] \geq E[N(k_0)] \quad \text{for all } k \in K(\alpha, \beta). \tag{5}$$

Theorem 2.1 shows that, by choosing the minimal possible value of k, Kennedy's algorithm achieves the fastest mean convergence rate of $L_n(k)$ for any given parameter choice and the tolerance level Δ. In addition, since at each stage the algorithm requires the fewest random numbers and least computation for sorting the generated values, the expected generation time can be minimized as well. We refer to this optimized version of Kennedy's algorithm as "Kennedy's Minimal-K" algorithm (or simply "Kennedy's MK" algorithm), and its steps are summarized as follows.

Kennedy's MK Algorithm

Step 1. Solve for k_0 and the corresponding values of p, q, r.

Step 2. Let $n = 0$, $k = k_0$, $A_0 = 0$, $B_0 = 1$, $\Delta = \Delta_0$.

Step 3. Generate k independent *Uniform*$(0, 1)$ variables and denote their sorted values by $U_{(1)}, \ldots, U_{(k)}$.

Step 4. Generate a random number U.
If $U \leq p$, set $A_{n+1} = U_{(1)}$, $B_{n+1} = B_n$.
If $p < U \leq p + q$, set $A_{n+1} = A_n$, $B_{n+1} = U_{(k)}$.
Otherwise, set $A_{n+1} = U_{(1)}$, $B_{n+1} = U_{(k)}$.

Step 5. If $(B_{n+1} - A_{n+1}) \leq \Delta_0$, then $x = (B_{n+1} + A_{n+1})/2$, and stop.
If $U \leq p$, generate U' from *Uniform*(A_{n+1}, B_{n+1}); sort $U', U_{(2)}, \ldots, U_{(k)}$ and denote the sorted values by $U_{(1)}, \ldots, U_{(k)}$.
If $p < U \leq p + q$, generate U' from *Uniform*(A_{n+1}, B_{n+1}); sort $U', U_{(1)}, \ldots, U_{(k-1)}$ and denote the sorted values by $U_{(1)}, \ldots, U_{(k)}$.
Otherwise, generate independent U' and U'' from *Uniform*(A_{n+1}, B_{n+1}); sort $U', U'', U_{(2)}, \ldots, U_{(k-1)}$ and denote the sorted values by $U_{(1)}, \ldots, U_{(k)}$.
Set $n = n + 1$, and go to Step 4.

Performance Evaluation

In this section, we first sketch the possible values of the shape parameters for the following well-known algorithms for beta generation: BN by Ahrens and Dieter, simple rejection (SR), order statistics method (OS), Jöhnk's method (JK), general switching algorithm (GS) by Atkinson and Whittaker, Forsythe's method, Cheng's BA/BB/BC, Sakasegawa's B00/B01/B11, B4PE by Schmeiser and Babu, and BPRS by Zechner and Stadlober. We then evaluate Kennedy's MK algorithm by comparing its performance with the state-of-the-art algorithms (from those listed above) in terms of the following performance measures: (i) computer generation time; (ii) program set-up time; (iii) goodness of fit; and (v) amount of random number generation required. For practical purposes, in some comparisons we also include the result of the default beta random number generators for three popular statistical software packages R, SAS, and Matlab (the latest version). Note that all performance measures were obtained based on a large number of simulation trials that were executed on 2GHz Pentium 4 processors with 1GB of cache. Computer programs were all written in Fortran, where a 32-bit linear congruential random number generator, the Quicksort and insertion sort algorithms (for sorting generated values) were used.

Valid Choices of (α, β)

The possible values of (α, β) for different beta generation algorithms are shown in Fig. 2. In comparison with Fig. 1, we see that Kennedy's MK algorithm can generate beta variates with a fairly wide range of (α, β). But on the other hand, Jöhnk's method, general switching algorithm, Forsythe's method, and Cheng's BA method can generate beta variates for all $\alpha, \beta > 0$.

Computer Generation Time

From the simulation results of the computer generation time with the previous comparative studies done by Sakasegawa (1983) and Zechner and Stadlober (1993), we present the following guideline for choosing the beta algorithm with the fastest generation speed:

- For $\alpha, \beta < 1$, choose Kennedy's MK algorithm if $\alpha + \beta > 1.2$; otherwise, choose Sakasegawa's B00 algorithm.
- For $\alpha < 1 < \beta$ or $\alpha > 1 > \beta$, choose Sakasegawa's B01 algorithm.
- For $\alpha, \beta > 1$, choose Schmeiser and Babu's B4PE algorithm if one parameter is close to 1 and the other is large (say >4); otherwise, choose Zechner and Stadlober's BPRS algorithm.

Goodness of Fit

Another way of evaluating different algorithms is to examine their generated random points and see how close is the empirical distribution function to the corresponding exact beta distribution. This can be done through the Kolmogorov–Smirnov statistic

$$D = \underset{0 \le x \le 1}{\text{Maximum}} \, |F_e(x) - F(x)|, \tag{18}$$

where $F_e(x)$ denotes the empirical distribution function computed from the generated random points $X_1, \ldots, X_n$, viz., $F_e(x) = \frac{1}{n}\sum_{i=1}^{n} I\{X_i \le x\}$, and $F(x)$ denotes the true beta distribution function. Note that the smaller the quantity D is, the better the underlying generating mechanism is in terms of goodness of fit.

The simulation results show that, none of the algorithms discussed seem to perform poorly in terms of goodness of fit.

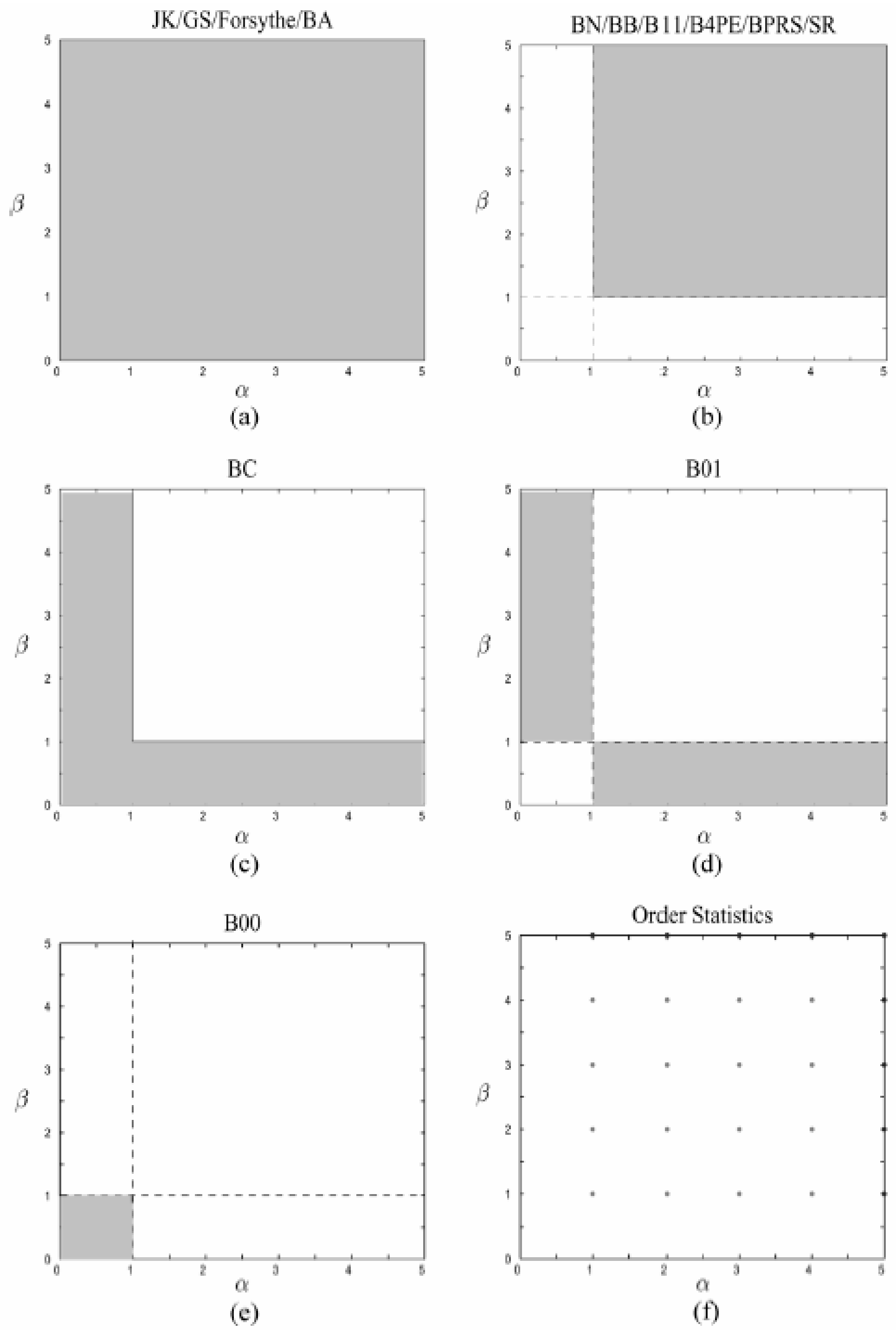

Figure 2. The possible range of (α, β) (grey part) for different beta algorithms. Note that the values on the dash lines are not possible.

Amount of Random Number Generation Required

In this section, we compare all the algorithms (not including the statistical software packages) through their amounts of random numbers required for each generation of beta variate. Intuitively, the fewer random numbers required for each generation, the more likely the algorithm can produce consecutively "independent" beta variates.

Based on the simulation results, we present the following guideline for choosing the beta generation algorithm that minimizes the amount of random number generation required:

- For $\alpha, \beta < 1$, choose Jöhnk's method if $\alpha + \beta < 1$; otherwise, choose Sakasegawa's B00 algorithm.
- For $\alpha < 1 < \beta$ or $\alpha > 1 > \beta$, choose Sakasegawa's B01 algorithm.
- For $\alpha, \beta > 1$, choose Zechner and Stadlober's BPRS algorithm.

(五) 計畫成果自評

The findings of this work can be divided into two parts. In the first part, we identified all parameter choices of beta variables that can possibly be generated by Kennedy's algorithm. We have further shown that an optimal parameter setting in Kennedy's algorithm, referred to as Kennedy's MK algorithm, can achieve the fastest speed of generation for any given valid parameter choice. In the second part, we have evaluated a class of well-known beta algorithms by means of various performance measures. The empirical study carried out shows that: (i) the proposed Kennedy's MK algorithm is faster than all existing algorithms for generating beta variates when $\alpha < 1$, $\beta < 1$, and $1.2 < \alpha + \beta < 2$; (ii) none of the beta algorithms considered in this study perform poorly in terms of goodness of fit; and (iii) the amount of random number generation required is in line with the algorithm's computer time, especially for algorithms based on rejection methods. From the numerical results, we present three guidelines for choosing the best suited beta algorithm. We believe that these guidelines will be quite useful for anyone requiring to perform a large beta random number generation. For example, the second guideline can be utilized to generate the Dirichlet random vectors, where all elements can be obtained by consecutively generating independent beta variates with different shape parameters (see algorithm DIR-3 by Narayanan, 1990). We are currently examining this approach and we hope to report the findings in a future paper.

The result of this study has been published in ***Communications in Statistics – Simulation and Computation*, 38: pp.750-770, 2009.**

(六) 參考文獻

Ahrens, J. H., Dieter, U. (1974). Computer methods for sampling from gamma, beta, Poisson and binomial distributions. *Computing (Vienna)* 12:223–246.

Arnold, B. C., Balakrishnan, N., Nagaraja, H. N. (1992). *A First Course in Order Statistics*. New York: John Wiley & Sons.

Atkinson, A. C. (1979). A family of switching algorithms for the computer generation of beta random variables. *Biometrika* 66:141–145.

Atkinson, A. C., Pearce, M. C. (1976). The computer generation of beta, gamma and normal random variables. *Journal of the Royal Statistical Society*, Series A, 139:297–312.

Atkinson, A. C., Whittaker, J. (1976). A switching algorithm for the generation of beta random variables with at lease one parameter less than 1. *Journal of the Royal Statistical Society*, Series A, 139:462–467.

Cheng, R. C. H. (1978). Generating beta variates with nonintegral shape parameters. *Communications of the Association for Computing Machinery* 21:317–322.

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

Forsythe, G. E. (1972). von Neumann's comparison method for random sampling from the normal and other distributions. *Mathematics of Computation* 26:817–826.

Gupta, A. K., Nadarajah, S. (2004). *Handbook of Beta Distribution and Its Applications*. New York: Marcel Dekker.

Jöhnk, M. D. (1964). Erzeugung von betaverteilten und gammaverteilten zuffallszahlen. *Metrika* 8:5–15.

Johnson, N. L., Kotz, S. (1990). Use of moments in deriving distributions and some characterizations. *Mathematical Scientist* 15:42–52.

Johnson, N. L., Kotz, S., Balakrishnan, N. (1995). *Continuous Univariate Distributions*. Vol. 2, 2nd ed. New York: John Wiley & Sons.

Kennedy, D. P. (1988). A note on stochastic search methods for global optimization. *Advances in Applied Probability* 20:476–478.

Morgan, M. G., Henrion, M. (1990). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge: Cambridge University Press.

Narayanan, A. (1990). Computer generation of Dirichlet random vectors. *Journal of Statistical Computation and Simulation* 36:19–30.

Sakasegawa, H. (1983). Stratified rejection and squeeze method for generating beta random numbers. *Annals of the Institute of Statistical Mathematics Part B*, 35:291–302.

Schmeiser, B. W., Babu, A. J. G. (1980). Beta variate generation via exponential majorizing functions. *Operations Research* 28:917–926.

Schmeiser, B. W., Babu, A. J. G. (1983). Errata. *Operations Research* 31:802.

Schmeiser, B. W., Shalaby, M. A. (1980). Acceptance/rejection methods for beta variate generation. *Journal of the American Statistical Association* 75:673–678.

Zechner, H., Stadlober, E. (1993). Generating beta variates via patchwork rejection. *Computing* 50:1–18.

出席國際學術會議心得報告

計畫編號	NSC 97-2118-M-004-004
計畫名稱	貝他分配演算法之評估
出國人員姓名 服務機關及職稱	洪英超 中央大學統計研究所助理教授(97/8/1~98/7/31) 政治大學統計系助理教授(98/8/1~)
會議時間地點	時間：2009年8月1~6日 地點：Washington, DC, USA.
會議名稱	Annual Joint Statistical Meetings (ASA)
發表論文題目	Uniform Design over Convex Input Domains with Applications to Computer Experiments

一、參加會議經過

2009 ASA Joint Statistical Meeting於2009年8月1~6日假Washington DC的Convention Center舉行。JSM是統計學家北美洲的最大年度聚會。它是由American Statistical Association、the Institute of Mathematical Statistics、the International Biometric Society (ENAR and WNAR)和the Statistical Society of Canada聯合舉辦，今年超過四千人參加。會議的內容包括oral presentations, panel sessions, poster presentations, continuing education courses, exhibit hall (with state-of-the-art statistical products and opportunities), placement service, society和section business meetings, committee meetings, social activities,和networking opportunities。本人提出的方法受到多方的重視。除此之外，我也得到許多與會人士的建議與回饋。

二、與會心得

本次與會的人士皆來自高科技的先進國家(如美國、英國、德國)。所以我們也感到與世界強國競爭的壓力。我想臺灣若要在各領域佔有一席之地，當務之急必須投下更多的人力與設備，並儘可能鼓勵年輕學者多參加大型國際會議。如此才能掌握研究的趨勢與動脈，做到真正的國際化與國際交流。