Variance reduction in Monte Carlo evaluation of residual connectedness network reliability

Héctor Cancela - **María E. Urquhart**

cancela, **urquhart@fing.edu.uy**

Operations Research Department

Engineering School

University of the Republic of Uruguay

Contents

1. Residual connectedness network reliability model
2. Standard Monte Carlo method
3. Recursive variance reduction (RVR) method
4. RVR properties
5. Numerical illustration
6. Conclusions

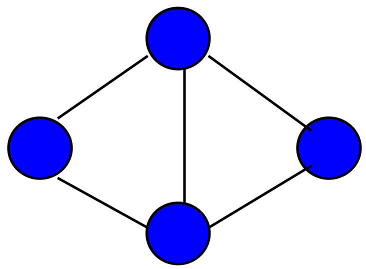
A Model

- Communication network:

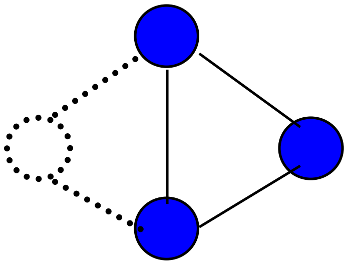
  undirected graph $\boldsymbol{G = (V,E)}$

  - **perfect lines (without failures)**
  - **nodes subject to independent failures**
  - **static model**
  - **operation probability of**

    node $u \in V$ is r_u



Network residual connectedness reliability



- Reliability measure $R(G)$:
  - $\tilde{G}$: subgraph of G induced by operational nodes
  - $Y(G)$: random variable "state of network G"

$$Y(G)=\begin{cases}1 & \text{if } \tilde{G} \text{ connected}\\ 0 & \text{otherwise}\end{cases}$$

  - Network residual connectedness reliability:

$$R(G)=E\{Y(G)\}=P\left\{\tilde{G}\text{ is connected}\right\}$$

Standard Monte Carlo Method (MCM)

- Unbiased estimator of $R(G)$

$$\hat{Y}(G) = \frac{1}{N}\sum_{i=1}^{N} Y^{(i)} \quad ; \quad Y^{(i)} \text{ is r.v. i.i.d. } Y(G)$$

- Variance

$$Var\{\hat{Y}(G)\} = \frac{Var\{Y(G)\}}{N} = \frac{R(G)\left(1-R(G)\right)}{N}$$

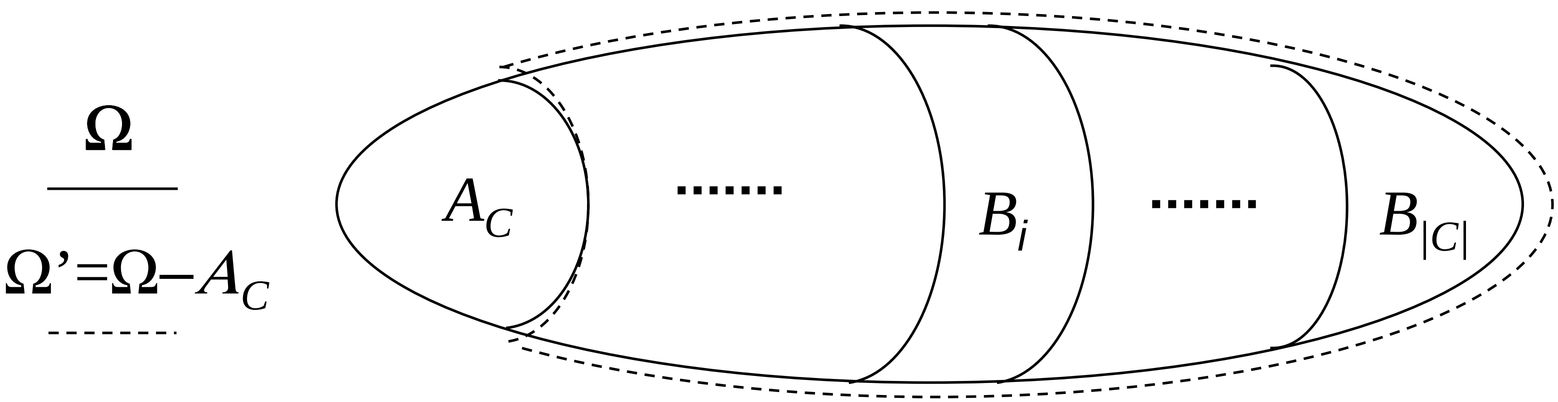
- Level ε confidence interval (applying CLT, with N big enough)

$$\hat{Y} \pm \xi\sqrt{\frac{Var\{Y(G)\}}{N}} \text{ with } \xi = G^{-1}\left(1-\frac{\varepsilon}{2}\right), \; G(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt$$

Recursive Variance Reduction (RVR) - preliminary definitions

- $C = \{u \in \mathrm{N} \,/\, r_u < 1\} = \{u_1, \ldots, u_{|C|}\}$ **: set of non-perfect nodes**
- A_C **: event "all nodes of C are operational"**
- B_i**: event "nodes** $u_1, \ldots, u_{i-1}$ **are operational, node** u_i **fails"**

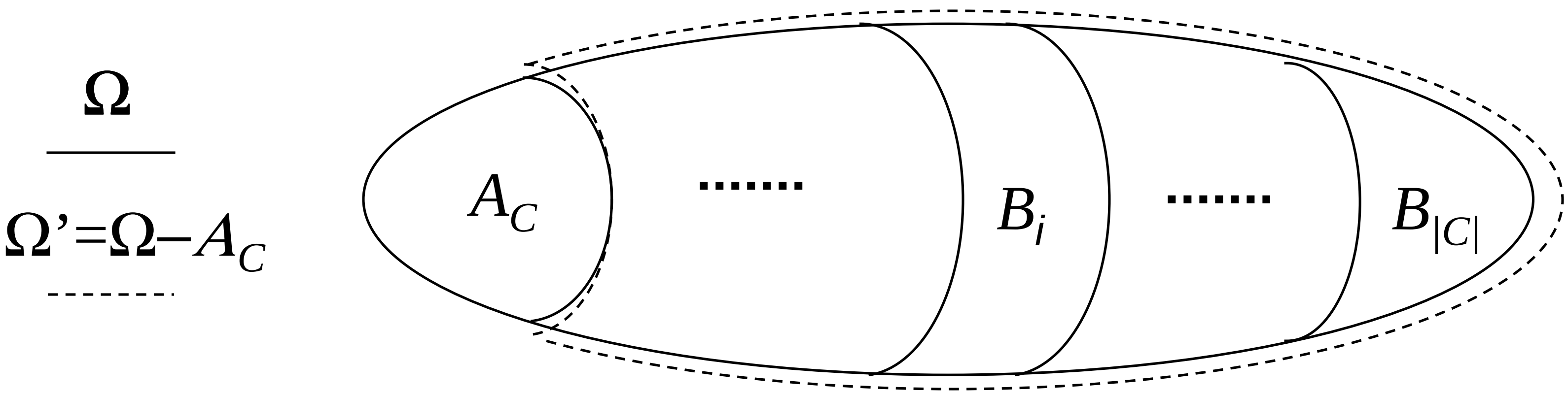
  $1 <= i <= |C|$



Recursive Variance Reduction (RVR) - preliminary definitions

- B_i: event "nodes $u_1,\ldots,u_{i-1}$ are operational, node u_i fails"
- G_i : network $(G/u_1\ldots/u_{i-1}) - u_i$, $1 \le i \le |C|$
- W: r.v. with distribution:

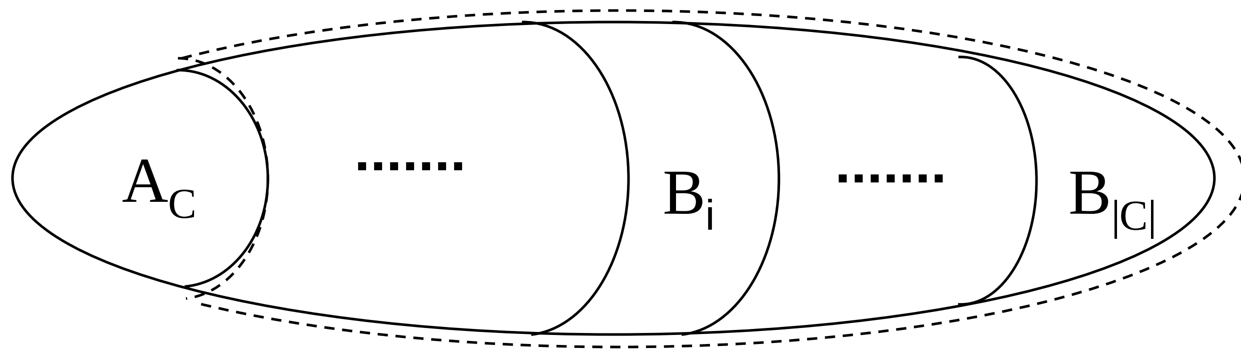
$$P\{W=i\} = \frac{P\{B_i\}}{(1-P\{A_C\})} = \frac{P\{B_i\}}{P\{\Omega'\}}$$



RVR estimator

Ω ———

$\Omega' = \Omega - A_C$ - - - - - - -



- Exact reliability

$$R(G) = \begin{cases} 1 \times P(A_C) + \sum_i R(G_i) P(B_i) & \textbf{if } G \textbf{ is connected} \\ 0 \times P(A_C) + \sum_i R(G_i) P(B_i) & \textbf{if } G \textbf{ is not connected} \end{cases}$$

- RVR estimator

$$Z(G) = \begin{cases} P(A_C) + P(\Omega') \sum_i Z(G_i)\, IND_{(W=i)} & \textbf{if } G \textbf{ is connected} \\ P(\Omega') \sum_i Z(G_i)\, IND_{(W=i)} & \textbf{if } G \textbf{ is not connected} \end{cases}$$

W r.v. with distribution $P\{W = i\} = \dfrac{P\{B_i\}}{(1 - P\{A_C\})} = \dfrac{P\{B_i\}}{P\{\Omega'\}}$

Recursive Variance Reduction RVR- Properties

- Property:
  - Z(G) is an unbiased estimator of R(G):

$$E(Z(G)) = R(G)$$

- Property:
  - Z(G) is more accurate than standard Monte Carlo

$$Var(Z(G)) \leq R(G)(1 - R(G)) = Var(Y(G))$$

- Relative accuracy index (equiv. efficiency)

$$S = Var(Y(G))/Var(Z(G))$$

RVR method

⇒ Input : network G

⇒ Output: a value of r.v. $Z(G)$

⇒ Pseudocode:

- End recursion conditions:
  - If all nodes are perfect and G connected : Return (1)
  - If all nodes are perfect and G not connected: Return (0)
- Preliminary computations:
  - Find C=set of non-perfect nodes
  - Generate a trial of W with distribution $P\{W = i\}=P\{Bi\}/P\{\Omega'\}$
  - Build network $G_i =(G /u_1.../u_{i-1})-u_i$
- Recursive call:
  - If G connected: Return $(P\{A_C\}+ P\{\Omega'\}.\mathrm{RVR}(G_i))$
  - Si G not connected Return ($P\{\Omega'\}.\mathrm{RVR}(G_i)$)

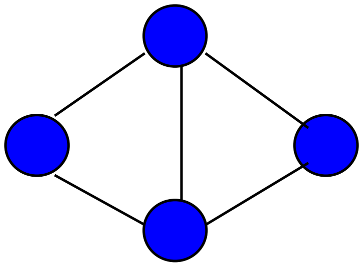
RVR method
Computational complexity

- Property:
  - complexity of a single RVR call: O(max(|V|,|*E*|))
  - maximum recursion depth: |V|
- Property:
  - if |V|=*n*, and all nodes have reliability *r*, the average recursion depth is bounded by

$$Depth(n) \le \sum_{k=1}^{n} \left(1 \Big/ \sum_{i=0}^{k-1} r^i \right)$$

  - Unreliable networks: if r→0, Depth(n) →n
  - Very reliable networks: if r→1, Depth(n) →H_n~ log(n)

Numerical illustration

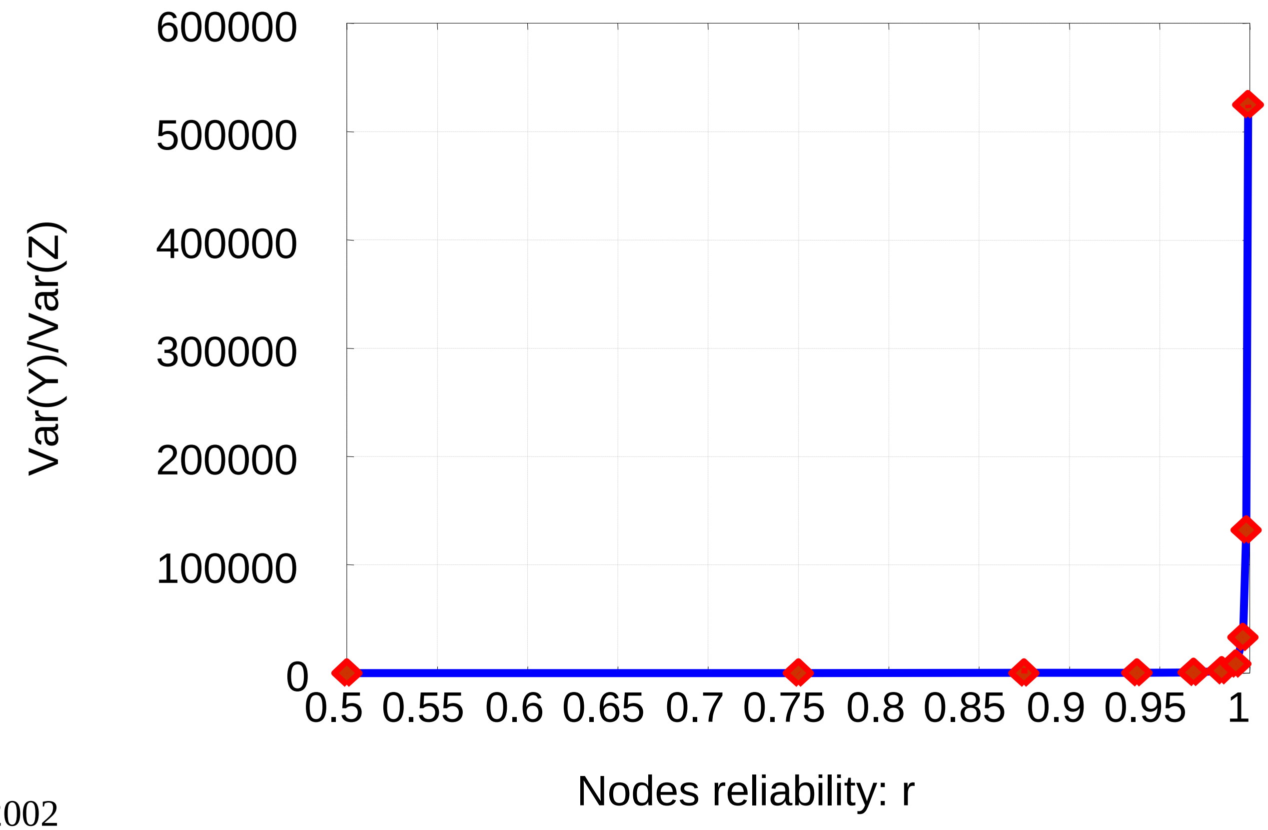


- “Bridge” network
- Node reliability $r=1-2^{-k}$, k=1,2,...,10
- Results:

k	r	R(G)	S=Var(Y)/Var(Z)
1	0.5	0.6875000	4.4
2	0.75	0.9023438	12.3
3	0.875	0.9724121	40.3
4	0.9375	0.9926605	144
5	0.96875	0.9981070	544
6	0.984375	0.9995193	2110
7	0.9921875	0.9998789	8320
8	0.99609375	0.9999696	33000
9	0.998046875	0.9999924	132000
10	0.9990234375	0.9999981	525000

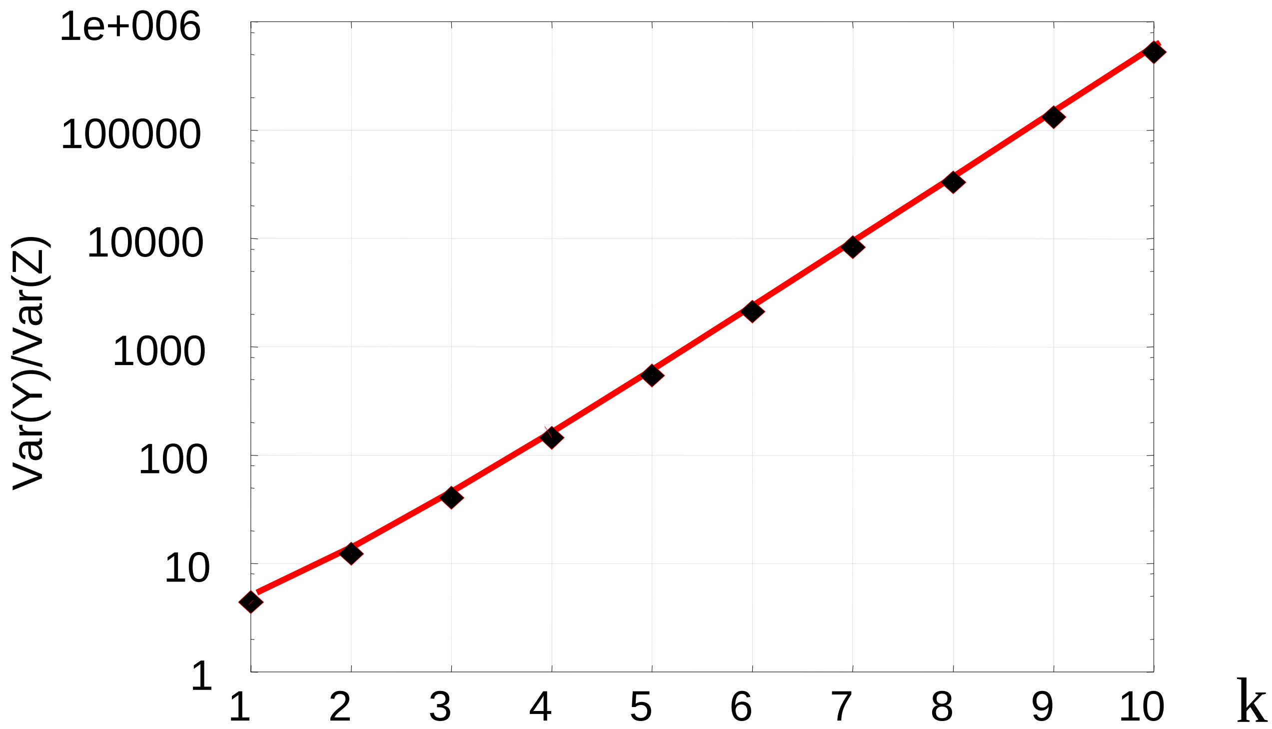
Relative efficiency index or speed-up behavior

- Relative precision S=Var(Y)/Var(Z) as function of $r=1-2^{-k}$ (usual scales)



Relative efficiency index behavior

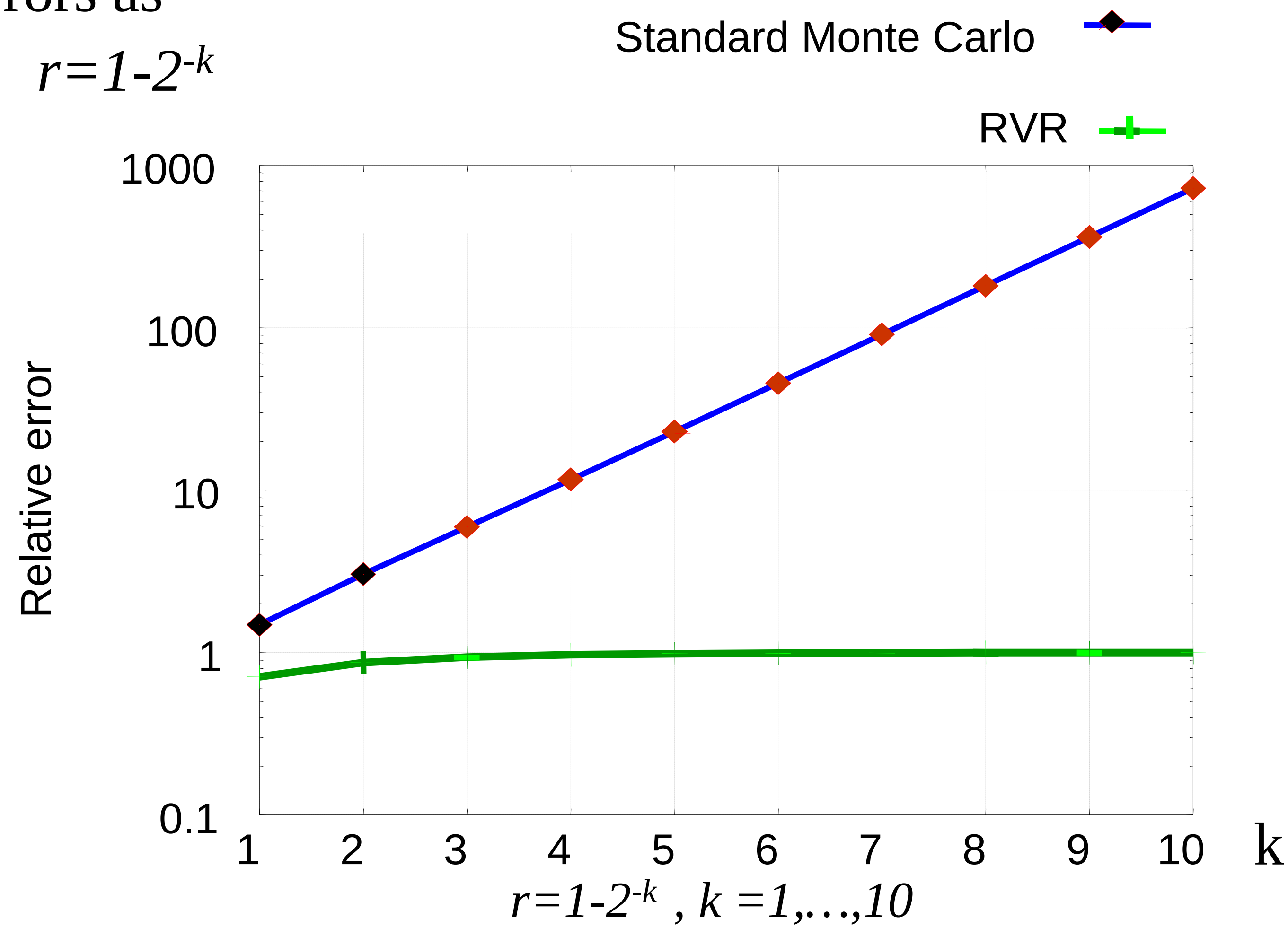
- Relative precision S=Var(Y)/Var(Z) as function of $r=1-2^{-k}$ (logarithmic scales on both axes)



$r=1-2^{-k}$, $k=1,\ldots,10$

Relative error comparison

Relative errors as function of $r=1-2^{-k}$



Conclusions

- RVR method is useful for reliability evaluation (in particular, residual connectivity).
- Specially adapted for very reliable systems:
  - Smaller computational complexity.
  - Greater precision.
- Numerical example suggests stable relative error.
- Future work:
  - Theoretical analysis of relative error behavior
  - Study of recursion depth for non-identical nodes.

References

1. H. Cancela, M. Urquhart *"Adapting RVR Simulation Techniques for Residual Connectedness Network Reliability Models"*, IEEE Transactions on Computers, vol 51 no 4**, pp** 439-443, 2002.
2. H.Cancela, M.El. Khadiri, *"A recursive variance-reduction algorithm for estimating communicaction network reliability"*, IEEE Trans. Reliability, vol 44, no 4, pp 595-602, 1995.
3. K. Sutner, A. Satyanarayana, and C. Suffel. *"The Complexity of the residual node connectedness reliability problem"*, SIAM J. Computing, vol 20, no 1, pp 149-155, 1991.
4. F. Boesch, A. Satyanarayana, and C. Suffel, *"On residual connectedness network reliability"* DIMAC Series in Discrete Math and Theoretical Computer Science, vol 5, pp 51-59, 1991.