JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

ELSEVIER Journal of Computational and Applied Mathematics 64 (1995) 163–176

# Opportunities of robust regression for variance reduction in discrete event simulation

Gerrit K. Janssens*, Wim Deceuninck, Alex Van Breedam

*Faculteit TEW, University of Antwerp (RUCA), Middelheimlaan 1, B-2020 Antwerp, Belgium*

Received 9 November 1993

**Abstract**

Variance reduction increases the efficiency of discrete event simulation. The method of using control variates searches for variables with known mean which are correlated with the response variable. The more correlated they are, the more the variance of the response variable can be reduced to obtain a confidence interval. Mostly only a few replications are made. This enables outliers to give a false view on the reduction possibilities. To cope with this statistical problem, we propose the use of a robust regression method in determining optimal weights for the control variables.

## 1. Introduction

Discrete event simulation driven by random input produces random output. As large-scale simulations require large amounts of computer time and storage, statistical analysis of this output can become costly. The cost can be so high that the precision measured by confidence interval width can be very poor [12, Ch. 11].

As confidence intervals are obtained from the variance of the simulation output, there is a need for variance reduction. Variance reduction techniques (VRT) aim to replace the original sampling procedure by another one in order to obtain the same expected value but with a smaller variance. In the literature VRT are mostly used in estimating the mean value, and not in estimating variances, quantiles or serial correlation coefficients [9].

The most widely used VRT are in the class of correlation methods. This type can be classified into methods based on induced correlation and based on control variates. Induced correlation means that the experiment induces a positive or negative correlation between blocks of simulation runs by manipulating the input random numbers. Two very well known methods are: (1) the use of

* Corresponding author.


*SSDI* 0377-0427(95)00013-5

common random numbers to compare two or more alternatives; (2) antithetic variates to estimate the mean response of a system.

*Common random numbers* decrease the variance on the difference of the estimated value by making the covariance positive. The method seems to be efficient in very simple (queueing) models, but not applicable to complex simulation output [2, 24].

*Antithetic variates* are based on the fact that the variance of the response variable in various replications of the simulation run is small if the covariance between the responses is negative. Schruben and Margolin [20] remark that also this method is only suitable for simple models.

*Control variates* attempt to take advantage of correlation between random variables to obtain a variance reduction. The method has been introduced in [11].

In the following sections we briefly describe the method and we will warn the user of possible dangers. A proposal is made to avoid these pitfalls by using a robust regression method. The method is illustrated in various applications: data traffic in a local area network, vehicle traffic at a light-controlled crossroads, a closed-loop conveyor system with loading and unloading stations, and a single server queue with a generalized switched Poisson arrival process.

## 2. Applicability of control variates

Consider a random variable $C$ with known mean $\mu_c = E(C)$ serving as a control variate for the response variable $Y$. As an example the expected value of an interarrival time or of a service time can serve.

Looking for an estimator for $Y$, it can be seen that, for each constant value $b$

$$Y(b) = Y - b(C - \mu_c)$$

is an unbiased estimator of $\mu = E(Y)$. As

$$\mathrm{Var}[Y(b)] = \mathrm{Var}[Y] - 2b\,\mathrm{Cov}[Y, C] + b^2\,\mathrm{Var}[C]$$

there exists a value of $b$, $\beta$, which minimizes $\mathrm{Var}[Y(b)]$. This value is:

$$\beta = \frac{\mathrm{Cov}[Y, C]}{\mathrm{Var}[C]}$$

from which

$$\mathrm{Var}[Y(b)] = (1 - \rho_{YC}^2)\cdot\mathrm{Var}[Y],$$

where $\rho_{YC}$ is the correlation coefficient between $Y$ and $C$.

The more correlated $C$ is with the estimator $Y$, the greater the reduction in variance.

The development illustrated with one control variate can be extended to the case of more than one. As in the case of a single control variate, the optimal weights must be estimated. The estimates of the optimal weights turn out to be identical to least-squares estimates of the coefficients in a certain linear regression model.

For a complex model a large number of possible control variates exist. It is not a good idea to use them all as variance is associated with the need to estimate the optimal coefficients [12, Ch. 11].

Control variates should be chosen to have a high correlation with the response variable, and, if possible, have low variance.

The need for variance reduction is highest in cases where the variability of the response variable is high, e.g., in heavy traffic situations and in transient situations. As an illustration the following example can serve. A local area network with 8 stations has a star configuration and uses the Hubnet protocol (cf. infra). Packets arrive according to a Poisson process with arrival intensity 1/100 μs. The packet length is 53 bytes. On a 50 MHz network this requires a delay of 8.5 μs. The network diameter, combined with the retry time is 32 bits (corresponding to 0.64 μs). Five replications have been run, starting from an empty-and-idle situation, and waiting and system times have been registered after 250, 500, 750 and 1000 packets sent. As control variate the measured interarrival time is used. Table 1 shows significance ($t$-values) of the coefficient of the variable "realized interarrival time" and explanation ($R^2$) for the linear regressions.

This example tells us that the use of the interarrival time as control variate can be efficient, but that its efficiency can be highly different. In its initial phase, where the system warms up, it plays an important role. The more the system reaches its steady state, its contribution becomes weaker. Then, however, the variance of the response variable is lower and the need for reduction becomes smaller.

Due to budget limitations, the number of replications is kept low in practice. By this, regression is made on a small number of observations. The user should be extremely careful with the interpretation of the $t$-values and the regression's explanatory power. The coefficients are in this case extremely sensitive to outliers.

It has been suggested in [4] that a more robust regression can be a help in reducing this kind of misinterpretation. In his simulation model [3] for the Hubnet protocol, the "least median of squares" (LMS)-method developed in [19] is used. The technique estimates regression coefficients based on a LMS-criterion (instead of the "least sum of squares"-criterion used in OLS) as an aid in identifying outliers. The purified sample afterwards is used in a re-weighted OLS-model. It is not required to eliminate the outliers in determining a confidence interval for the response variable. An outlier, detected by LMS, only indicates it cannot serve in variance reduction.

Table 1
Regression characteristics for variance reduction of the waiting time using the interarrival time as control variable

| | | $t$ | $R^2$ |
|---|---|---|---|
| 250 | Waiting time | − 7.1 | 0.92 |
| | System time | − 6.9 | 0.92 |
| 500 | Waiting time | − 2.5 | 0.57 |
| | System time | − 1.9 | 0.41 |
| 750 | Waiting time | − 3.6 | 0.75 |
| | System time | − 3.0 | 0.66 |
| 1000 | Waiting time | − 3.0 | 0.66 |
| | System time | − 3.4 | 0.73 |

## 3. Application 1: Simulation of the Hubnet protocol

In local area networks access control is distributed among the sending stations. They share a common medium for transmission and, by this, collision between messages can occur. These collisions destroy the information content of the packets. Therefore it can be expected that, as offered load increases, also the number of collisions increases and the throughput degrades. Therefore collision-avoidance local area networks have been developed with different topologies. We discuss the performance of such a LAN with a star topology called Hubnet [13].

In order to build a performance model for Hubnet, one has to understand in detail the working of the protocol. The Hubnet protocol consists of two parts: one at the nodes and one at the Hub (the central node of the star). An "idle" node can transmit a packet from its transmission buffer if (1) it has sensed the echo of its previous packet, i.e., it has been transmitted successfully, or (2) at least one round-trip propagation delay has passed since last transmission, i.e., the packet has not been transmitted. The node remains in a "ready" state; afterwards it turns again into the "idle" state. The hub can also be in two states: the "idle" state and the "busy" state. A packet which finds the hub in the "idle" state acquires the hub and puts it in the "busy" state. All further arriving packets are discarded. The packet is broadcast on all outgoing links and, by this, will also be received by its originator. Analytical approximative performance models for Hubnet have been developed in [7, 5]. For the latter a discrete event simulation served as a validation tool for the analytical model.

The parameter set for the simulation model is: the number of transmitting stations, the packet length, the round-trip propagation delay, and the parameter of the arrival process. If it is assumed that all stations transmit according to a Poisson process with the same arrival intensity, then the only stochastic part in the simulation is the interarrival time. Let $R_n$ be the system time, i.e., the total time spent in a station queue, retrying a number of times, and the deterministic propagation and transmission time, of the $n$th packet departing the system. We are interested in estimating the mean of the steady-state distribution to which the distribution of the $R_n$ converges. A reasonable estimator of this mean is the average of the first $N$ system times:

$$Y = \frac{1}{N} \sum_{n=1}^{N} R_n.$$

For these first $N$ packets, we can consider the average interarrival time

$$C = \frac{1}{N} \sum_{n=1}^{N} T_n$$

knowing that $\{T_n,\ n = 1,2,3, \dots\}$ is the random input sequence of i.i.d. interarrival times. The expectation of $C$ is the known mean interarrival time. The variable $C$ is a reasonable control variable as it can be expected that $C$ is negatively correlated with $Y$. If, in a simulation run, the average interarrival time is larger than its expected value, the average system time tends to be less than its expected value.

The method is illustrated by means of the waiting time example after 500 packets sent from Table 1. The arithmetic mean computed from five simulation runs equals 3.70 μs (stand. dev. 0.59 μs). The regression equation obtained by LMS is:

$$\text{Waiting time} = 11.95\ \mu\text{s} + 0.682 \times \text{Interarrival time}.$$

By this equation we try to obtain new values for the observations. The expected Waiting time should occur from the regression equation by using the expected Interarrival time. If the expected value is not known or data are collected in periods during which the simulation has not yet reached the steady-state, the average value (here 13.25 μs) can be taken.

The Interarrival time is split into two parts: an expected value part (13.25 μs) and a surplus to obtain:

$$\text{Measured Waiting Time} + 0.682 \times \text{Surplus} = 11.95\ \mu\text{s} - 0.682 \times 13.25 + \text{Error Term}.$$

The right hand side of the previous equation equals the new value of the observation. By this, we obtain five modified values from which an average value and a standard deviation can be calculated in order to define a confidence interval. The latter variance is smaller than the original.

If the contribution of the measured average interarrival time in the regression is not significant in the regression, we check for outliers and recalculate the regression with the outliers removed. If this reweighted regression has a significant coefficient for the average interarrival times, new values for the nonoutliers observations are obtained using the above mentioned method. In this case as expected value part the median of the observed values is taken (and not the average value as above, due to the presence of outliers).

If, however, the reweighted regression does not have a significant coefficient for the average interarrival time we conclude that the variable cannot lead to any variance reduction and the original sample is left.

To conclude our example Table 2 shows the variation coefficient (in %) of the regression coefficient before and after the reduction, be it by means of OLS, or by means of the robust LMS method.

Table 2
Variation coefficients for different experiments before and after reduction using the interarrival time as control variable

| | | Before | After OLS | After LMS |
|---|---|---|---|---|
| 250 | Waiting time | 25.2 | 5.9 | |
| | System time | 10.6 | 2.6 | |
| 500 | Waiting time | 15.9 | 8.7 | |
| | System time | 4.5 | | 3.2 |
| 750 | Waiting time | 19.5 | 8.2 | |
| | System time | 5.1 | 2.5 | |
| 1000 | Waiting time | 8.5 | 4.3 | |
| | System time | 6.6 | 2.0 | |

## 4. Application 2: Simulation of a light-controlled traffic intersection

Microscopic models of vehicle flow are models in which vehicles are considered as separate entities. Assumptions are made on the behavior of the individual user. To model the behavior at a cross-roads use can be made of queueing models. Our aim is to analyze the delays at an intersection controlled by a fixed-cycle traffic signal. A simple version of this problem is to find the delay to a single stream of traffic that forms an input to the intersection. To build a simulation model specification is required of: (1) the signal discipline, (2) the arrival process for incoming vehicles, and (3) the process describing the manner in which vehicles pass through the intersection.

The queueing discipline can be considered FIFO. Only priority cars could change this. As this type of car does not appear under standard conditions, we will not consider this case. We will consider the arrivals to follow a Poisson process. This is a realistic approximation if traffic is light. In cases where interaction between vehicles cannot be neglected, e.g., a compound Poisson process can be used [15].

The intersection considered is controlled by traffic lights. Both horizontal and vertical directions are unidirectional. Both roads can have multiple lanes. A simulation model written in SIMAN is added in Appendix A. Both the model file and the experiment file are presented. The behavior of a driver is modeled as follows: the driver crosses the intersection if (1) the traffic light in his direction shows green, and (2) a lane is free on the intersection.

The experiments are run with deterministic crossing times. Car arrival processes are Poisson processes with different arrival rates. As dependent performance variables the system times in both directions are chosen. Both variables are dependent on the realized arrival rates in both directions. A multiple regression can be made in this case. The LMS-method requires a number of observations at least double the number of coefficients to be estimated. In our case, this number equals three: the regression constant and the regression coefficients for both realized arrival rates.

From the experiments we learned that using two explanatory variables increases the variance reduction power only in cases where the utilization of the intersection is high in both directions. If the simulation budget is limited, we can advice to use only the mean interarrival time of the higher utilization direction as control variable.

## 5. Application 3: Simulation of a closed-loop conveyor system

From an operational point of view conveyors can be classified into four categories: closed-discrete, closed-continuous, line-discrete and line-continuous [18]. A conveyor is closed if it moves in a closed loop, otherwise it has a begin and an end in which case it is called a line conveyor. A conveyor is of the continuous type if it can be loaded on every part. It is of the discrete type if it consists of equidistantly spaced carriers or hooks connected to some moving chain.

We will discuss the closed-loop discrete system. Little theoretical results are known because the underlying stochastic processes are very complicated. All investigations are done by simulation studies. The only results available, under very strict assumptions, can be found in [16, 17].

The model of the conveyor system under study is as follows: units arrive according to a renewal process where they are stored in a queue with infinite capacity. The units are transported to a number of unloading stations by one or more carriers attached to a closed chain moving at a constant speed. When a carrier passes the queue a loading station immediately fills the carrier with as many units as possible. The unloading stations have a buffer with finite capacity. Unloading takes place immediately. If not enough places are available to unload, the rest of the units circulates further searching for empty places at another unloading station. To each unloading station a server is connected, processing the units for some random time, after which they leave the system.

Nawijn [18] proposes the following Kendall notation for the system:

$$A|(N, M)|B|s|K,$$

in which
$A(.)$ is the interarrival time distribution at the loading station, $B(.)$ the processing time distribution, $N$ the number of carriers, $M$ the capacity of each carrier, $s$ the number of unloading stations, $K - 1$ the storage capacity at each unloading station.
The example worked out uses the following parameters for the model:
$A(.)$ is the exponential distribution with $\lambda = 4$ units/time unit, $B(.)$ the exponential distribution with $\mu = 1$ unit/time unit, $N = 10$ carriers, $M = 5$ units/carrier, $s = 5$ unloading stations, $K - 1 = +\infty$.
The SIMAN code for this simulation model can be found in Appendix B.

This is an example in which control variates can fail. A performance measure as system time can be controlled by the arrival process. But as the service discipline is so complex we can hardly expect that changes in the mean interarrival time will have great impact on the performance measure. Experiments with the above mentioned model also showed that using the mean service time is of no use.

Each of the five stations has its own mean service time and due to the complex interplay, the aggregate mean service time is not able to reduce the variance. The only way to reduce variance can be found in using all mean service times individually as explanatory variables. This method however should require at least 12 observations (coefficients for the five service times and for the constant in the regression). In these cases robust regression will be of little help.

## 6. Application 4: Simulation of a generalized switched Poisson process

The generalized switched Poisson process results from an alternate switching between two Poisson arrival processes, characterized by the arrival rates $\lambda_1$ and $\lambda_2$. The times that the arrival generating process stays in each of both states are called phase length times. Those times form a process of independent identically distributed random variables denoted by $T_1$ and $T_2$. Following the notation of Tran-Gia [21], the class of Generalized Switched Poisson Processes is denoted by SPP($G_1$,$G_2$) where $G_1$ and $G_2$ denote the distribution of the phase lengths $T_1$ and $T_2$.

The SPP($G_1$, $G_2$)-process can be completely characterized by the four random variables (r.v):
$T_1$ r.v. for the length of phase 1 (mean $= 1/\omega_1$),
$T_2$ r.v. for the length of phase 2 (mean $= 1/\omega_2$),
$T_{A_1}$ r.v. for the interarrival time during phase 1 (mean $= 1/\lambda_1$),
$T_{A_2}$ r.v. for the interarrival time during phase 2 (mean $= 1/\lambda_2$).

As the process alternates between two Poisson processes, the r.v.'s $T_{A_1}$ and $T_{A_2}$ are exponentially distributed:

$$F_{A_i}(t) = \Pr\{T_{A_i} \leqslant t\} = 1 - \mathrm{e}^{-\lambda_i t}, \quad i = 1, 2, \qquad E[T_{A_i}] = \frac{1}{\lambda_i}, \quad i = 1, 2.$$

We consider single server queueing systems with infinite capacity and with service time distribution $G$, for which the SPP($G_1$, $G_2$)-process serves as the input process.

The SPP($G_1$, $G_2$)-process in which both phase lengths are exponentially distributed are denoted by the SPP($M$, $M$)-process. In this case the process is a two-state Markov modulated process [7].

The SPP($M$, $M$) is not a renewal process. The autocorrelation coefficient of the sequence of interarrival intervals is given in [23]. From that result it can be learned that the only cases in which the process is a renewal process are the cases where $\lambda_1 \cdot \lambda_2 = 0$ and $\lambda_1 = \lambda_2$. The first case is called an interrupted Poisson process (IPP). The second case is the simple Poisson process.

The cases in which one of both arrival rates $\lambda_1$ and $\lambda_2$ equals zero form another class of SPP($G_1$, $G_2$)-systems, in case the Interrupted Poisson Process (IPP). We denote the input process as IPP($G_1$, $G_2$). The IPP($G$, $M$) is a renewal process.

The simulation for this model is not coded in SIMAN. Most simulation packages have problems in generating nonstationary Poisson processes. The ad hoc methods, which are used often, differ in the way they treat the generation of the interarrival times which overlap two time intervals with a different arrival rate. Three methods are proposed [8]: (1) use the parameter of the time interval in which the overlapping interarrival time *begins*; (2) use the parameter of the time interval in which the overlapping interarrival time *ends*; (3) the overlapping interarrival time is *discarded* and the next arrival time is set equal to the begin time of the next time interval.

A "correct" approach is proposed in [10]. It has been found in [22] that the latter approach is the only one to be used in simulations. This approach is included in the simulator, which has been documented earlier [6]. The procedure generating correct switched Poisson arrivals is included in Appendix C. Note that not only the arrival process is difficult in a nonstationary case, but also the statistical analysis. Advice on this topic can be found in [14].

To illustrate the case in which many control variables are available, we use the SPP ($G_1$, $G_2$)/G/1-queue, with the following characteristics:

—phase lengths are exponential,
—service times are exponential,
—queue capacity is infinite.

Simulations are run with various values of: the number of events in a period, the ratio of phase length and overload factor. Average arrival rates and service rates are chosen to cover the whole range of utilization factors of the queue.

As potential candidates for controlling variance the following measures can serve: utilization factor, mean service time, variance of the service time, variation coefficient of the service time, mean interarrival time, variance of the interarrival time, variation coefficient of the interarrival time, ratio of the phase lengths, overload factor, and autocorrelation coefficient between successive interarrival times.

One hundred replications have been run for each experiment to decrease the influence of a possible outlier. By using stepwise regression a subset of candidates is selected: only four came out to be significant for the complete set of experiments. Utilization appears in all experiments; Mean Service Time in 80% of the cases; Mean Arrival Time in 60% of the cases and Variance of the Service Time in 15% of the cases.

## 7. Conclusion

When applying control variates as a variance reduction technique (VRT), a random variable $C$, correlated with the response variable $Y$, is used. Since OLS regression can be sensitive to outliers, especially with a small number of simulation replications, robust regression should be used.

In the Hubnet protocol simulation results, reductions ranging from 29 to 77% are obtained. In the traffic intersection tests, using two explanatory variables is only useful in high-utilization situations. This is conform with the increased need for variance reduction when the response variable has high variability.

In the $4|1|10|5|5| + \infty$ (Kendall notation) closed-loop discrete conveyor system, control variates cannot be expected to be useful because of the service discipline complexity. Besides, robust regression will be of little help because of a relatively large number of observations needed.

Finally, looking for useful control variates in the GSPP/G/1-queue simulation, utilization and mean service time, and to a less powerful extent, mean arrival time and service time variance, turned out to be significant.

From the various and multidisciplinary applications, it can be seen that control variates can be a very useful VRT, especially because invalid conclusions (if outliers are likely) can be avoided by using robust regression.

## Appendix A

This appendix contains the SIMAN code for a light-controlled intersection. Comments on the logic of the model are given in block sequences starting with ";".

```
BEGIN;
;
     CREATE:EX(1,1):MARK(1);
;       Cars arrive according to a Poisson process (parameter values can be found in
:       parameter set 1 in the experiment file)
     QUEUE,1;
;       Cars wait in line for light direction 1. The number of available resources from
;       'LIGHT1' represent the number of available lanes on the intersection. This
;       number is set to zero if the light shows red.
     SEIZE:LIGHT1;
;       Start crossing the intersection
     DELAY:CO(3,1);
;       Cross the intersection during a deterministic time, of which the value is given
;       in parameter set 3 in the experiment file
```

```
    RELEASE:LIGHT1;
;      Leave the intersection
    TALLY:1,INT(1):DISPOSE;
;      Register the time interval between now and the arrival in the system (see
;      MARK(1) in the CREATE block above) and let the car leave the system
;      (DISPOSE)
;
    CREATE:EX(2, 1):MARK(1);
    QUEUE,2;
    SEIZE:LICHT2;
    DELAY:CO(3,1);
    TALLY:2,INT(1):DISPOSE;
END;
```

An example of the SIMAN experiment file corresponding to the previous model file is:

```
BEGIN;
PROJECT,INTERSECTION2,GKJ,04/11/1994;
DISCRETE,400,1,2;
TALLIES:1,TIME IN SYSTEM 1:2,TIME IN SYSTEM 2;
RESOURCES:1,LIGHT1,SCHED(1):2,LIGHT2,SCHED(2);
SCHEDULES:1,0*2.25,2*4:2,1*2,0*4.25;
PARAMETERS:1.2:2,4:3.1;
REPLICATE, 1,0,100;
DSTAT:1,NQ(1),QUEUE 1:2,NQ(2), QUEUE 2:3,NR(1), UTIL 1:4,NR(2),UTIL 2;
END;
```

## Appendix B

This appendix contains the SIMAN code for a closed-loop conveyor model. Comments on the logic of the model are given in block seqeunces starting with ";".

```
BEGIN
    CREATE:EX(1,1):MARK(1);
    TALLY:6,BET(1);
    COUNT:6;
    STATION,6;
    ASSIGN:A(1) = 1;
    QUEUE,6,1800,LOST;
    ACCES:BUCKET,1;
    CONVEY:BUCKET,A(1);
;
CYCLE   STATION,1-5;
    BRANCH,1:
      IF, NR(M).EQ.0,PROC:
      ELSE,CONT;
```

```
;
PROC   EXIT:BUCKET,1;
     QUEUE,M;
     SEIZE:MACHINE(M):MARK(2);
     DELAY:EX(2,1);
     RELEASE:MACHINE(M);
     TALLY:M,INT(2);
     COUNT:DISPOSE;
;
CONT   BRANCH,1:
     IF,M.EQ.5,INIT:
     ELSE,VOLG;
;
INIT   ASSIGN:A(1) = 1;
     COUNT:8;
     CONVEY:BUCKET,A(1);
;
VOLG   ASSIGN:A(1) = A(1) + 1;
     CONVEY:BUCKET,A(1);
;
LOST   COUNT:7:DISPOSE;
END;
```

An example of the experiment file for this model is:

```
BEGIN;
  PROJECT,CONVEYOR,GKJ,04/17/1994;
  D ISCRETE,1850,3,8,6;
  REPLICATE,1,0,1000;
  PARAMETERS:10.5:
     2,1;
  RESOURCES:1-5,MACHINE;
  SEGMENTS:1,6,1-55,2-10,3-10,4-10,5-10,6-55;
  CONVEYORS:1,BUCKET,1,60,15,1A;
  COUNTERS:1,PROC. ON MACH1:
     2,PROC. ON MACH2:
     3,PROC. ON MACH3:
     4,PROC. ON MACH4:
     5,PROC. ON MACH5:
     6,TOTAL IN SYSTEM:
     7, LOST:
     8,NUMBER RECIRC;
  TALLIES:1,SERVICE MACH1:
     2,SERVICE MACH2:
     3,SERVICE MACH3:
     4,SERVICE MACH4:
```

```
    5,SERVICE MACH5:
    6,INTERARRIVAL TIME:
  DSTAT:1,NR(1),MACH1 UTIL.:
    2,NR(2),MACH2 UTIL.:
    3,NR(3),MACH3 UTIL.:
    4,NR(4),MACH4 UTIL.:
    5,NR(5),MACH5 UTIL:
    6,LC(1),CONVEYOR UTIL.:
    7,NQ(6),NUMBER WAITING;
END;
```

**Appendix C**

```
Function Next_Arrival_Time (Last_Arrival_Time:real):real;
var art_rand, dummy_random, help_phase_length:real;
  i:integer;
Function Compute_Next_Arrival_Time (phase:integer; randno:real):real;
begin
  Compute_Next_Arrival_Time: = Last_Arrival_Time-(ln(1-randno)/Lambda[phase]);
end; (* computer_next_arrival_time *)
Function Compute_Random (phase:integer):real;
const ignore_value = 15.0;
var diff : real ;
begin
    diff : = New_Upper_bound - Last_Arrival_time;
    if (Lambda[phase]* diff < = ignore_value) then
      Compute_Random := 1-(exp(-(Lambda[phase]*diff)))
    else
      Compute_Random := 1.0;
end; (* compute_random *)
begin (* Next_Arrival_Time *)
  art_rand : = Compute_Random(phase) ;
  dummy_random:= random;
while dummy_random > art_rand do
    begin
    dummy_random := (dummy_random-art_rand)/(1-art_rand);
      Last_Arrival_Time := New_Upper_bound;
      case phase of
        1:phase := 2;
        2:begin
          phase := 1;
          Count_periods := Count_periods + 1
        end;
```

```
        end;
        help_phase_length := phase_length(phase);
        paper1_phase_length (help_phase_length, phase);
        New_Upper_bound := New_Upper_bound + help_phase_length;
        art_rand := Compute_Random(phase);
      end;
    Next_Arrival_Time : = Compute_Next_Arrival_Time(phase,dummy_random);
  end; (* next_arrival_time *)
```

## References


[1] H. Heffes, A class of data traffic process – covariance function characterization and related queueing results, *Bell System Technical J.* **59** (1980) 897–929.

[2] R.G. Heikes, D.C. Montgomery and R.L. Rardin, Using common random numbers in simulation experiments, *Simulation* **25** (1976) 81–85.

[3] G.K. Janssens, A performance model for a conflict-free access method to a tree-structured local area network, in: H.H. Adelsberger and F. Broeckx, Eds., *Discrete Event Simulation and Operations Research* (Society for Computer Simulation, Ghent/San Diego, 1987) XXIII–XXVI.

[4] G.K. Janssens, Analysis and Simulation of the performance of a conflict-free access protocol for local data communication supporting the optimal design of the interconnection of networks of different types, Ph.D. Thesis (in Dutch), Free University of Brussels, 1989.

[5] G.K. Janssens, Collision dependent performance model for a star topology local area network, in: O. Balci, R. Sharda and S.A. Zenios, Eds., *Computer Science and Operations Research: New Development in their Interfaces* (Pergamon Press, Oxford, 1992) 117–126.

[6] G.K. Janssens and A. Van Breedam, A simulator for single server queues with a generalized switched Poisson process, RUCA/STE/Working Paper 91/04, University of Antwerp, Belgium, 1991.

[7] A.E. Kamal and V.C. Hamacher, Analysis of a star local area network with collision avoidance, *Proc. of INFOCOM* 86 (1991) 546–555.

[8] F.C. Kaminsky and D.L. Rumpf, Simulating nonstationary Poisson processes: a comparison of alternatives including the correct approach, *Simulation* **29** (1) (1977) 17–20.

[9] J.P.C. Kleijnen, *Statistical Techniques in Simulation* (Marcel Dekker, New York, 1979).

[10] R.W. Klein and S.D. Roberts, A time-varying Poisson arrival process generator, *Simulation* **43** (4) (1984) 193–195.

[11] S.S. Lavenberg and P.D. Welch, A perspective on the use of control variables to increase the efficiency of Monte Carlo simulations, *Management Science* **27** (3) (1981) 322–335.

[12] A.M. Law and W.D. Kelton *Simulation Modeling and Analysis* (McGraw-Hill, New York, 2nd ed., 1991).

[13] E.S. Lee and P.I.P. Boulton, The principles and performance of Hubnet: a 50 Mb/s fiber optics local area network, *IEEE J. Selected Areas in Commun.* **SAC-1** (1983) 711–720.

[14] P.A.W. Lewis and G.S. Shedler, Statistical analysis of non-stationary events in a data base system, *IBM J. Res. Dev.* **20** (1976) 465–482.

[15] D.R. McNeill and G.H. Weiss, Delay problems for isolated intersections, in: D.C. Gazis, Ed., *Traffic Science* (Wiley, New York, 1974) 109–174.

[16] W.T. Morris, *Analysis for Materials Handling Management* (Irwin, Homewood, IL, 1962).

[17] E.J. Muth, Modelling and systems analysis of multi-station closed-loop conveyors, *Internat. J. Prod. Res.* **13** (1975) 559–566.

[18] W.M. Nawijn, Stochastic conveyor systems, Ph.D. Thesis, University of Twente, 1983.

[19] P.J. Rousseeuw, Least median of squares regression, *J. Amer. Statist. Assoc.* **79** (1984) 871–880.

[20] L.W. Schruben and Margolin, Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments, *J. Amer. Statist. Assoc.* **73** (1978) 504–520.

[21] P. Tran-Gia, A class of renewal interrupted Poisson processes and applications to queueing systems, *Zeitschrift fur Oper. Res.* **32** (1988) 231–250.

[22] A. Van Breedam and F. Broeckx, Using simulation for generating non-stationary Poisson processes, in: G.C. Vansteenkiste et al., Eds., *Proc. 1990 European Simulation Symposium*, November 8–10, 1990, Ghent, Belgium (Society for Computer Simulation, 1990) 169–173.

[23] M.H. Van Hoorn and L.P. Seelen, The SPP/G/1 queue: a single server queue with a switched Poisson process as input process, *OR Spektrum* (1983) 207–218.

[24] R.D. Wright and T.E. Ramsay, On the effectiveness of common random numbers, *Management Science* **25** (1979) 649–656.