Queuing network simulation output analysis and parallel execution mechanisms

Panajotis Katsaros

TRACS Visitor

katsaros@csd.auth.gr

Dept. of Informatics

Aristotle University of Thessaloniki

G R E E C E

Queuing network simulations I

USE: Performance analysis

Computer (HW/SW) systems

Communication systems

closed networks

mixed networks

open networks

OBJECTIVE: usually, estimation of the mean of a performance measure

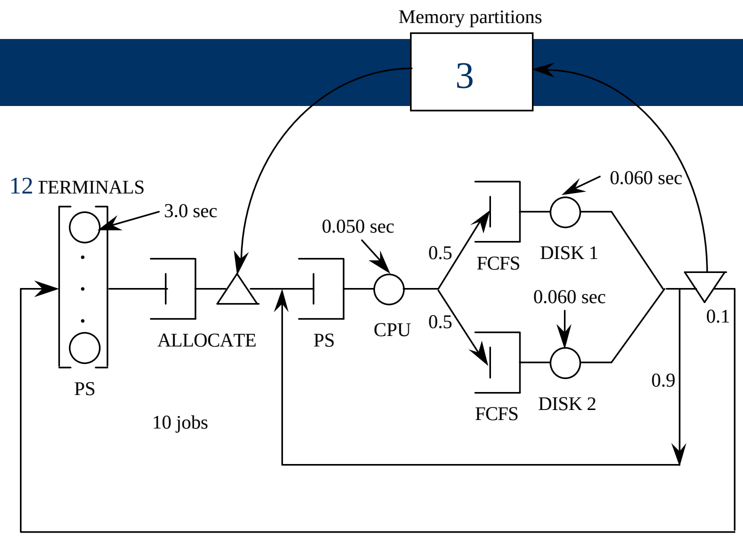
throughput, i.e. jobs served in the unit of time

response time

utilization, i.e. fraction of busy time

…… and the accuracy of the estimator

Queuing network simulations II



from the sequence of collected observations $x_1, x_2, \ldots, x_n$ an estimator of the mean μ_x can be

$$\overline{X}(n) = \sum_{i=1}^{n} \frac{x_i}{n}$$

and the accuracy of the estimate is assessed by the probability

$$P(\overline{X}(n) - \Delta_x \leq \mu_x \leq \overline{X}(n) - \Delta_x) = 1 - \alpha$$

which means that if the experiment were repeated a number of times, the interval $(\overline{X}(n) - \Delta_x, \overline{X}(n) + \Delta_x)$ would not contain the μ_x only in 100α% of cases

Queuing network simulation problems I

WHEN TO STOP?

- **Transient study**: the analyst is interested in performance measures over some relatively short period of time. In such cases, the results may depend quite strongly on the initial conditions of the simulation.
- **Steady-state study**: the simulation must typically be run long enough so that the effects of the initial state on the performance measures of interest are negligible.

STEADY-STATE→PROBLEM: HOW LONG?

… in order to answer, we have to be able to estimate the effects of the bias introduced by the selection of the initial state of the system

Queuing network simulation problems II

IS THE ESTIMATION GOOD?

- The **bias** measures the systematic deviation of the estimator from the true value of the estimated parameter. Thus, in the case of $\overline{X}(n)$

$$Bias[\overline{X}(n)] = E[\overline{X}(n) - \mu_x]$$

- The **variance**, which measures the mean deviation of the estimator from its mean value,

$$\sigma^2[\overline{X}(n)] = E[\{\overline{X}(n) - E[\overline{X}(n)]\}^2]$$

- The **mean square error** (MSE) of the estimator, defined as

$$MSE[\overline{X}(n)] = E\{[\overline{X}(n) - \mu_x]^2\}$$

Methods for simulation output analysis I

- The method of independent replications
- The method of batch means
- The method of overlapping batch means
- The regenerative method
- The method based on spectral analysis
- The method based on autoregressive representation
- The method based on standardized time series

Methods for simulation output analysis II

PROBLEMS WITH MOST METHODS:

- The initial transient period has to be estimated and observations collected in this period to be discarded. **Ignoring the existence of this period can lead to a significant bias of the final results**.
- The observations collected are statistically dependent (correlated). Most methods result in a quite complex algorithm in order either **to weaken autocorrelations** among observations **or to exploit the correlated nature** of observations in the analysis of variance needed for determining confidence intervals.

The regenerative method I

- The system is initialized in an appropriate recurrent state, which can be considered that occurs in the steady-state.

- Data collection is performed at the entry times into this state (**regenerative state**). We say that a **regenerative cycle** is completed. Estimates of variance are then easily computed since the generated observations are independent and identically distributed.

Introduced by Cox & Smith [1961] and independently developed by Fishman [1974] and Crane & Iglehart [1974].

The regenerative method II

APPLICABILITY: In most queuing network models

Shedler, G., “Regenerative stochastic simulation”, California, Academic Press, 1983

Every discrete event simulation is a Generalized semi-Markov process (GSMP). **If there is at least one service center that sees only one job class or it is such that jobs of the lowest priority are subject to pre-emption**, then there is a set of states which possess the regenerative property.

Even if the above are not satisfied, if the underlying GSMP is a Harris recurrent process then the regenerative method is also applicable.

In the worst case,

Gunther, F.L. & Wolff, R.W. ”The almost regenerative method” In Operations Research Vol. 28, No. 2, 1980

Regenerative estimation

If our aim is to estimate $k(f) = E[f(X)]$

and we call $Z_k(f) = \int_{T_{k-1}}^{T_k} f(X(u)) \cdot du$

then it can be proved

,

$$k(N) = \frac{\bar{Z}(N)}{\bar{\tau}(N)}$$ (A), where $\bar{\tau}(N)$ the average cycle length

and a 100α% confidence interval for k(f) is given by

$$\left[k(N) - \frac{s(N) \cdot F^{-1}\left(\frac{1+a}{2}\right)}{\sqrt{N} \cdot \bar{\tau}(N)}, k(N) + \frac{s(N) \cdot F^{-1}\left(\frac{1+a}{2}\right)}{\sqrt{N} \cdot \bar{\tau}(N)} \right]$$ (B)

where

$$s^2(N) = s_{11}^2(N) - 2k(N)s_{12}^2(N) + (k(N))^2 s_{22}^2(N)$$

with

$$s_{12}^2(N) = \frac{1}{N-1} \sum_{k=1}^{N} (Z_k(f) - \bar{Z}(N))(\tau_\kappa - \bar{\tau}(N))$$

$$s_{22}^2(N) = \frac{1}{N-1} \sum_{k=1}^{N} (\tau_k - \bar{\tau}(N))^2$$

$$s_{11}^2(N) = \frac{1}{N-1} \sum_{k=1}^{N} (Z_k(f) - \bar{Z}(N))^2$$

Regenerative estimation: Problem

- It can be shown that although the estimator given in (A) is consistent, i.e.

$$P(k(N) \xrightarrow[N \to \infty]{} \mu_x) = 1$$

  it is a **biased estimator** of μ_x.

  Thus, although the initialization bias has been eliminated, a new source of systematic errors has been introduced by the special form of the regenerative estimator.

- Other estimators used with the regenerative method:

  the Fieller estimator, the Beale estimator, the Jackknife estimator and the Tin estimator

- Comparative studies shown that the Jackknife and the Tin estimators give much less biased results

Sequential control procedure

HOW MANY REGENERATIVE CYCLES NEED TO BE COMPLETED?

It depends on the required confidence interval width.

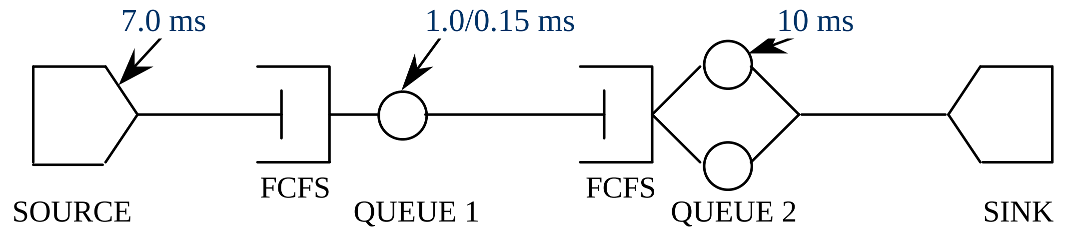
Generally, if a confidence interval of no more than 100δ% of k(f) half width is to be achieved, we need at least

$$N \geq \left(\frac{F^{-1}\left(\frac{1+a}{2}\right)}{\delta} \right)^2 \cdot \left(\frac{s(l)}{k(l) \cdot \overline{\tau}(l)} \right)^2$$

regenerative cycles.

However, even if the previous relation is satisfied a **normality test** has also to be applied, in order the estimated results to be valid.

Sample I: long simulation run



NAME: QUEUE 1

RESOURCE	UTILIZATION	THROUGHPUT	TOTAL LENGTH	RESPONSE TIME
ReqCIL	2 %	2 %	2 %	2 %
ActCIL	+/- 1.9 %	+/- 0.55 %	+/- 2 %	+/- 2 %
CYCLES	61	15	101268	99784
LBOUND	0.9329	0.14293	20.472	142.45
MEAN	0.95108	0.14372	20.889	145.35
UBOUND	0.96926	0.1445	21.307	148.26

NAME: QUEUE 2

RESOURCE	UTILIZATION	THROUGHPUT	TOTAL LENGTH	RESPONSE TIME
ReqCIL	2 %	2 %	2 %	2 %
ActCIL	+/- 2 %	+/- 0.55 %	+/- 2 %	+/- 2 %
CYCLES	66	15	1097	927
LBOUND	0.69578	0.14293	2.8189	19.71
MEAN	0.70992	0.14372	2.8762	20.11
UBOUND	0.72406	0.1445	2.9336	20.51

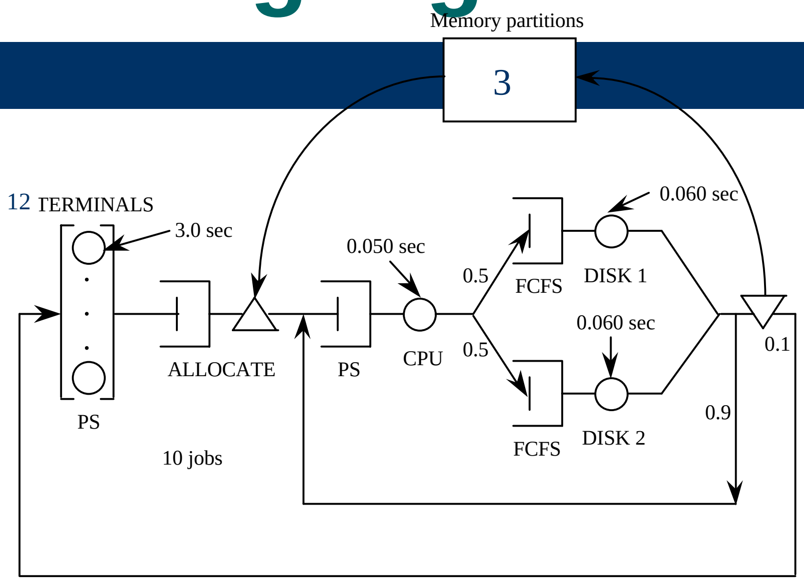
NUMBER OF EVENTS: 40415233
SIMULATED TIME 9.41485e+007
REQUIRED CYCLES 101268
NUMBER OF CYCLES: 101268
AVERAGE NUMBER OF EVENTS: 399
AVERAGE LENGTH: 929.696 C.I.:(915.951,943.442)

NUMBER OF CYCLES: 101268

AVERAGE NO OF EVENTS: 399

CPU TIME USE: 413.7 sec

Sample II: long regenerative cycles



NUMBER OF EVENTS: 1795297 NUMBER OF CYCLES: 260

SIMULATED TIME 54349.3 AVERAGE LENGTH: 209.036 C.I.:(181.383,236.689)

REQUIRED CYCLES 258.578

HALF CONFIDENCE INTERVAL LENGTH: 1% CYCLES: 260

AVERAGE NUMBER OF EVENTS: 6904

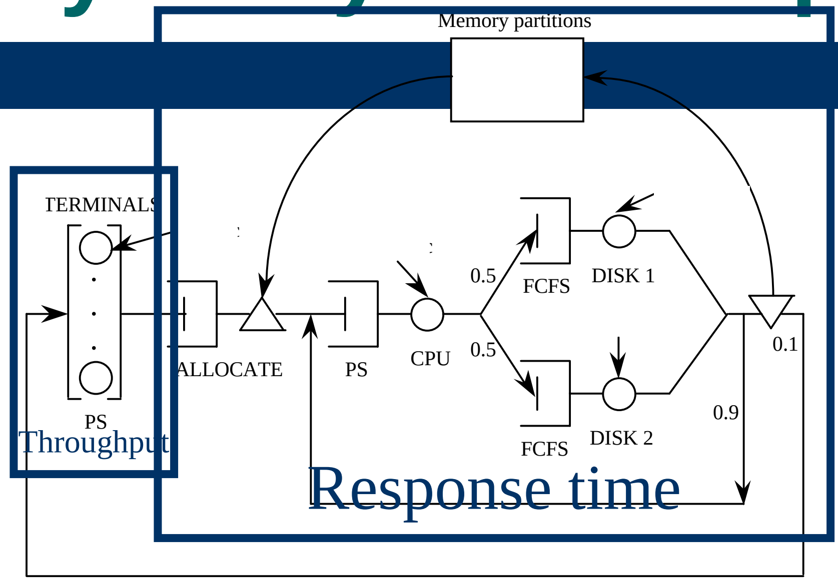
CPU TIME USE: 24.28 sec

Sensitivity analysis

Sensitivity analysis aims in understanding how the model responds to changes, in predicting and maybe optimizing it.

It requires a number of experimental data, which means that a sufficiently large number of simulation experiments with different input parameters has to take place.

Sensitivity analysis example



Experimental design: a balanced 3^{5-1} factorial design with 81 experiments=35 min CPU

The factors and the levels of the experiment were:

A: Number of terminals in 3 levels: 10, 25, 40

B: Memory partition in three levels: 2, 4, 6

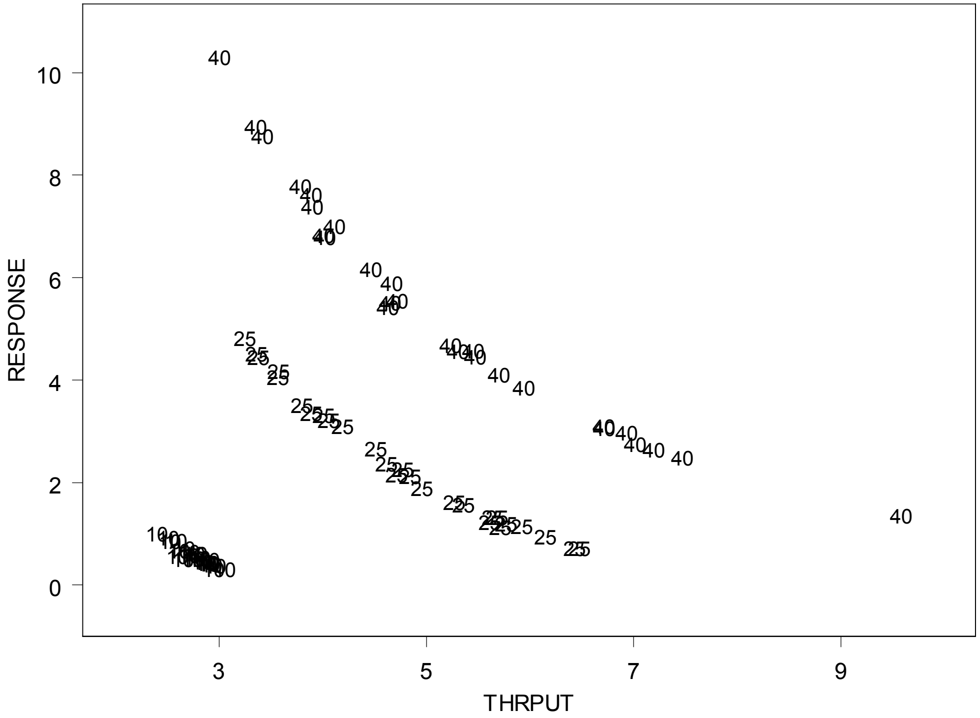
C: CPU speed in three levels: 0.009, 0.012, 0.015

D: Disk speed in three levels: 0.023, 0.028, 0.033

E: Number of disks in three levels: 1, 2, 3

243 cases

Sensitivity analysis example (cont.)



Sensitivity analysis example (cont.)

We used an advanced backward and stepwise regression procedure to find the metamodels that best fit to our data

Dependent Variable: LN(RESPONSE)

r-square=0.985

Coefficients

(Constant)	-2.031
A	0.146
B	-0.125
D	46.544
E	-0.800
A^2	-1.019E-03
B^2	2.727E-02
E^2	0.156
A*B	-3.306E-03
A*E	-5.128E-03
B*E	-6.563E-02
C*E	18.713
A^2	-1.019E-03
B^2	2.727E-02
E^2	0.156

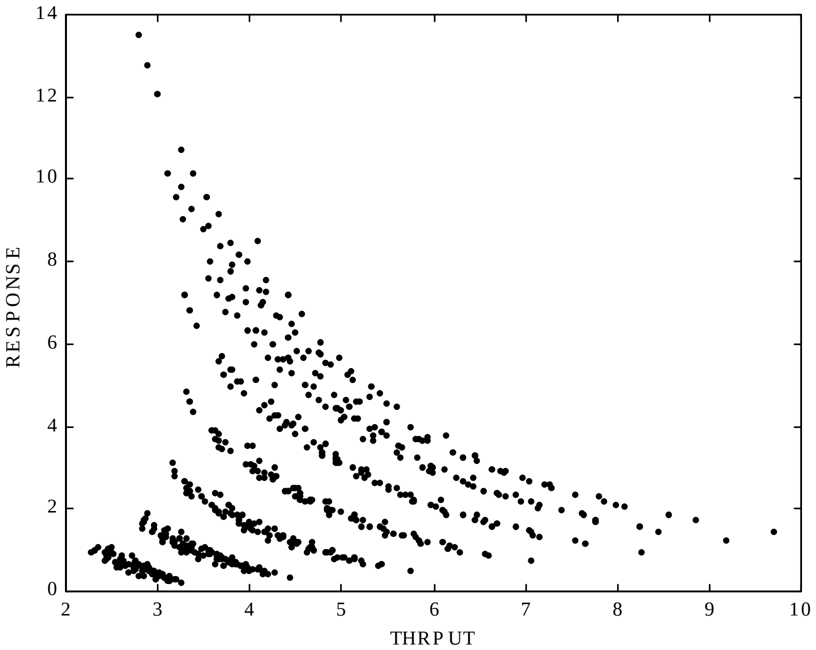
Dependent Variable: LN(THRPUT)

r-square=0.981

Coefficients

(Constant)	0.518
A	7.458E-02
D	-14.065
E	0.193
A^2	-1.023E-03
B^2	-8.498E-03
E^2	-8.235E-02
A*B	2.543E-03
A*C	-0.511
A*D	-0.706
A*E	6.156E-03
B*E	2.692E-02
C*D	499.882
C*E	-7.296
D*E	4.148

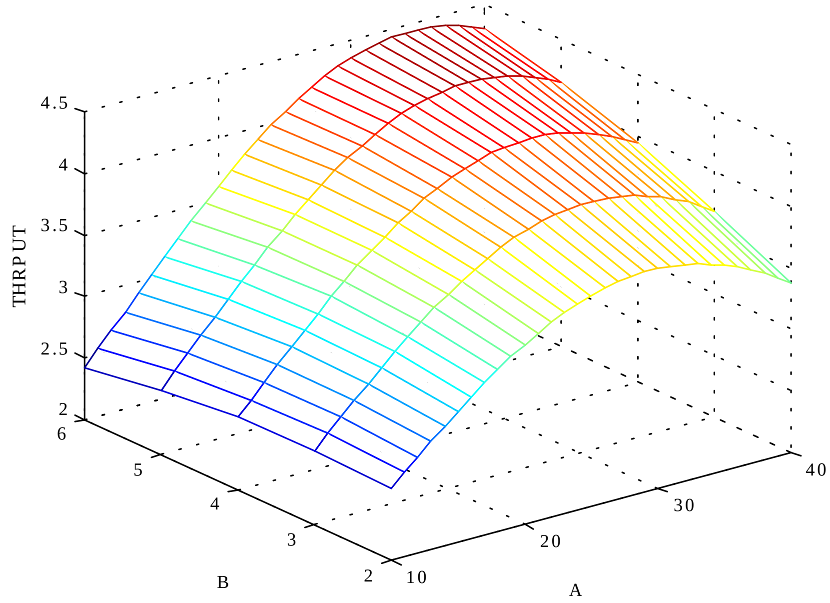
Sensitivity analysis example (cont.)



Predictions for different workloads:

10, 15, 20, 25, 30, 35, 40 terminals

Sensitivity analysis example (cont.)



Response surface for **THROUGHPUT** when

cpu speed: 0.012 disk speed: 0.028 n of disks: 1

Sensitivity analysis based on the use of likelihood ratios

- In this technique the occurrence of certain events are counted during the simulation. Then, the natural variation in the random process underlying the parameter is utilized, in order to produce the sensitivity estimate (derivative of expectations).
- ONLY APPLICABLE IN THE REGENERATIVE SIMULATION

Optimization by the use of likelihood ratios

The results are just noisy estimates of the real gradient values.

So, if our aim is system optimization an appropriate stochastic optimization algorithm (e.g. the Robbins-Monro algorithm) has to be applied.

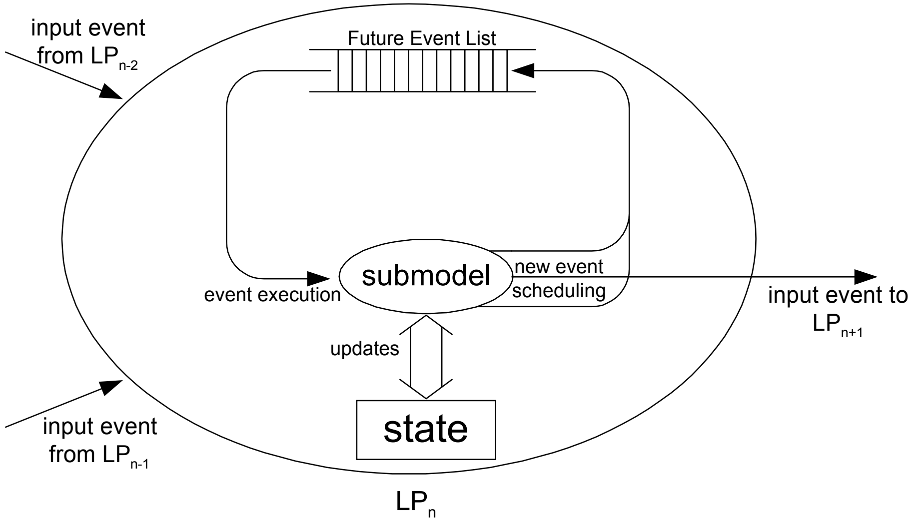
Parallel/distributed discrete event simulation

- Parallel/distributed discrete event simulation has been successfully used to overcome the problem of simulating complex and large models, but
- it has not been used yet in simulation output analysis
  - *known exceptions*: Raatikainen, K. ”Run Length Control using Parallel Spectral Methods” In Proceedings of the Winter Simulation Conference (December 1992)
  - Pawlikowski, K., Yau, V., McNickle,D., “Distributed and stochastic discrete-event simulation in parellel time streams” In Proc. of the Winter Simulation Conference (1994)

Parallel discrete event simulation

In the parallel discrete event simulation, the model is partitioned into a number of submodels which are called *logical processes* (LPs).

Thus, each LP is defined as a set of one or more queues and a Future Event List.



Parallel discrete event simulation: Chandy-Misra approach

In the Chandy-Misra execution mechanism, event processing adheres to the *local causality* constraint, which prescribes that events are processed in non decreasing timestamp order.

This execution mechanism has the potential for deadlock. For this reason it is usually applied together with an appropriate deadlock resolution scheme.

Parallel discrete event simulation: Time Warp approach

The Time Warp approach allows the occurrence of causality errors, but provides a mechanism to recover from them.

This assumes to keep open the possibility to roll back the LP to the most recently saved state and for this reason, each LP has to keep past state buffers, past input buffers, antimessage buffers etc.

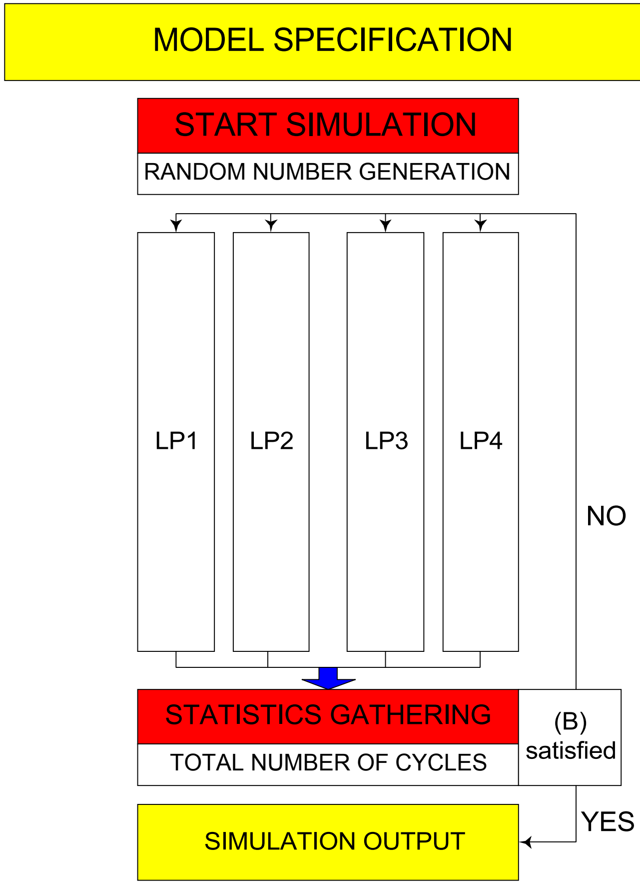
A memory management algorithm is usually applied in order to guarantee availability of a “sufficient” amount of memory.

Parallel regenerative queuing network simulation

Katsaros,P. and Lazos, C.,”Regenerative queuing network distributed simulation”, In. Proc. of the European Simulation Multiconference (2000)

- each LP contains the entire model
- there is no need for synchronization between the LPs: simulation time is not important, from the point of view that the experiment depends only on the return of the model to the same state, irrespective of the time instant that this will happen
- need for a termination algorithm that controls the execution of the different LPs: the sequential control algorithm applied

Parallel regenerative queuing network simulation depiction



Implementation considerations & aims

- Shared memory parallelization by the use of OpenMP and C++

AIMS:

  - To measure the succeeded speed-up and efficiency
  - To validate the accuracy of the obtained results (random sampling is being done in a different way in the parallel case)
  - Maybe to implement the single run sensitivity analysis method, which is based on the use of the likelihood ratios