DIRECT MODELING OF QUEUING SYSTEMS AND NETWORKS

Yury I. Ryzhikov[(a)], Vyacheslav A. Zelentsov[(a)], Semyon A. Potryasaev[(a,b)]

[(a)]Saint Petersburg Institute for Informatics and Automation of the RAS (SPIIRAS), St.Petersburg, Russia
[(b)]University ITMO, St.Petersburg, Russia

[(a)] ryzhbox@yandex.ru, [(b)]info@litsam.ru, [(c)]sokol@iias.spb.su

ABSTRACT
Queuing theory has numerous applications in the organization of production, analysis of information processes, transport problems, health care, military-technical tasks, etc. This report shows a clear lack of traditionally used GPSS simulation technology for the purposes indicated and features that are useful for direct imitation.

Keywords: queuing theory, simulation.

1. INTRODUCTION

In various areas of technology, in the organization of production and in the military there is a need to solve peculiar probability problems related to queuing systems of different types of requirements. This concept involves multiple recurrence of situations (many arrivals in and served requirements, a large number of using similar systems) and the statistical stability of the situation. The scope of its potential applications, in particular, include:

- in communications technology - the design of telephone and computing networks, analysis of communication protocols;
- in transport problems - analysis of traffic passing through a tunnel, the technical inspection of motor vehicles, the formation of trains, airport takeoff and landing, loading and unloading of ships at the sea and river ports;
- in the industry - the planning of assembly operations, flexible automated production, organization of equipment repair;
- in automated systems - evaluating waiting time of demands for computational work and preparation of data, processing the results of experiments, technology control;
- in health care - in determining the required number of hospital beds, stations and ambulances, doctors and nurses, needs in diagnostic and therapeutic equipment;
- in the judiciary - its staff size, capacity of corrective labor institutions, investigative staff headcount and operatives;
- in the field of science and education - the study of certain types of physical processes (registration of elementary particles, filtration, diffusion), launch the satellites, satellite data processing, calculation of the number of laboratory facilities; design and analysis of large libraries working;
- in the military - design of air-defense systems (each target can be regarded as "demand" on the service, i.e. firing), the organization of border security, based patrol outfits and in many other cases, such as those described above in relation to daily activities and combat troops .

Immediately, we emphasize that these types of problems have to be solved not only in the design of the newly created queuing networks and systems (QN, QS), but also in the available service - with increasing load; changing complexity of the processing; of failure, degradation or upgrading technology; reducing staff; revising requirements for expediting the processing of demands, etc.

2. SIMULATION MODELING IN GENERAL

Simulation modeling is the computer playing of the simulated system "life". It discusses its behavior by the set of algorithms for each type of event. The control program monitors the events preserving the sequence of their occurrence in modeling time, and then processes the accumulated statistics.

With the ability to sufficiently complete reality reflection (for example, a multi-stage service, inhomogeneous flows, channel locks because of the buffer limited capacity, a complex systems of priorities and other service disciplines, heterogeneous resources) simulation is applicable to problems irreducible to analytical and numerical methods, and complexity of the problem does not create any principal difficulties. Simulation is indispensable for testing new numerical and analytical techniques. In particular, author used it when developing approximate numerical methods for the analysis of systems with random selection from queue, of multi-level priority queues with a quantized service, as well as for systems and networks with "negative" demands and with impatient ones. The disadvantages of simulation are:

- high consumption of computer time;
- low accuracy of the probabilistic characteristics of rare events;
- difficulties of the system optimization (search of the optimum is conducting in the presence of random errors in the results);
- difficulty in obtaining general conclusions and recommendations.

Very popular belief that the problems of accuracy and optimization can be solved by increasing the number of tests. The table 1 shows his insolvency by calculating the number π using statistical tests – as a fraction of points in the circle inscribed in the unit square.

Table 1. Error in determining the number π by statistical tests

N	δ	N	δ	N	δ
1 000	9.8e-2	50 000	2.8e-3	2 000 000	-1.4e-4
2 000	8.6e-2	100 000	8.3e-3	5 000 000	-1.0e-5
5 000	5.6e-3	200 000	4.3e-3	10 000 000	2.9e-4
10 000	1.8e-2	500 000	4.3e-3	20 000 000	4.2e-5
20 000	2.2e-3	1 000 000	2.0e-3	50 000 000	1.2e-4

Simulation modeling on high level languages PL/1 and FORTRAN is very time consuming for the following reasons:

- due to the increasing complexity of the simulated systems the number of elements and monitored processes is constantly increasing;
- simulation models are difficult to debug because of the rich logic, mutual dependence of the algorithm branches, very time consuming runs and low accuracy of the results, in some cases able to hide an error;
- even minor changes in the model may require many corrections to the program and its debugging nearly over.

The solution may be using some standard schemes as models for general classes of systems. Modeling languages allow to write the simulation program in a form that resembles the description of the simulated system, and in such a way that small changes in the system behavior correspond to small changes in the program. It provides, having a minimum experience, fast writing and debugging of the simulation programs. The most known such systems are GPSS and its analogs. Undoubted advantages of modeling languages are:

- Simplification of the model description.
- Avoiding the programmer from many technical details.
- Automatic control of input program syntax correctness.

Established in the early 1960s. general-purpose simulation system GPSS to this day remains the most popular language in the world of simulation. It was used in the learning process by more than 100 Russian universities. GPSS program is a description of the trajectory of the message ("TRANSACT"). It contains the names and the order of used "facilities" (service devices) or "memories"; delays; logical conditions governing promotion of transacts; waypoints, where data collects about waiting time, etc.

The general rule of new software using is its testing on the problems with the known solution (preferably - analytic). For the M/M/1 system the theoretical queue length is $q = \rho^2 / (1-\rho) =$ 0.9^2 / (1-0.9) =8.1. Average waiting time w according to a Little's formula must be q / λ and in this case ($\lambda = 1$) is numerically equal to the average length of the queue. Theoretical probability of the idle system is $1-\rho = 0.1$. Therefore, on average a tenth of the applications should receive services without delay.

Table 2.Testing GPSS / W on the M/M/1 model

Index	Theory	Thousand of tests		
		50	200	500
Load factor	0.900	0.896	0.897	0.899
Average service time	0.900	0.899	0.902	0.901
Number of inputs		49843	199042	499032
Of them with zero expectation		5340	21001	50691
Average queue length	8.100	5.132	5.757	6.690
Average waiting time	8.100	5.148	5.785	6.703

Results associated with the expectation are improper. The reason for this can be only one: an insufficient quality of the pseudorandom numbers generator.

Finally, we discuss the temporal characteristics of the compared model realizations. To achieve the level of the timer 500 000 (and approximately the same number of served transacts) it was needed 187 seconds. The run of relevant FORTRAN program for the same raw data gave an average queue length 8.683 - much closer to the theoretical estimate, than the best of the results in Table 2. The run took only 6.26 seconds. Such a large difference is the natural price to pay for the versatility of GPSS interpreter.

The mentioned system is a very valuable tool simulation, free from the constraints of analytical and numerical methods, rather "transparent", allowing non-standard data processing and removing from the programmer the set of nontrivial problems of program writing and debugging. It has, however, several serious drawbacks:

1. The cumbersome of the system and excessive manifold primitives.
2. Lack of conceptual unity. In confirmation, we refer to the difference in treatment the elements of the matrix with a simple link and value changing.
3. Unsuccessful designations for relational operators L, G, E (it would be better to concord them with FORTRAN); SQR function is used for the square root (in Pascal this designate “square”); the state of logical keys is described as SET and RESET (literal translation - "installed" and "re-set") instead of ON, OFF; operand RE (the traditional meaning - the repetition of an action) means the removal, and it would be better to use DEL.
4. Prohibition to interrupt "the memories" (they can be used to model multi-service devices and, therefore, will be subject to interruptions).
5. Non-possibility to change the type and layout of graphs scales, as well as color and texture of the lines, making them indistinguishable from each other and/or the background in black-and-white output.
6. The argument of graphs may be the time only, so that the probability distribution of the system states can’t be presented automatically. By the way, this will necessarily require a logarithmic scale, the implementation of which is missing.
7. Opportunities of the formal optimization models are of questionable value because in real-world problems solving:
    - the number of channels in the network nodes is integer;
    - channels productivity is selected from a finite set of allowed values;
    - restrictions are imposed on the elements of a routing matrix numerous, reflecting processing technology.

3. DIRECT SIMULATION

Simulation environments solve the limited range of opportunities in principle, but due to excessive claims of the developers to universality they form the redundant programs with big time consuming. However, any simulation algorithm can be written on the usual procedural oriented programming language. Hence, there remains the need to write direct modeling programs on the languages such as modern FORTRAN. This applies in particular to learning in higher education, the purpose of which should be not the mastering of technologies, but the clarification of principles. Benefits of direct modeling are:

- no need for special software and its studying;
- «transparency» of model logic;
- the availability of built-in functions, library routines, statistical analysis tools, editing output and other features usually provided by a general-purpose programming systems;
- no restrictions on the model composition and logic;
- possibilities to control the experiment - in particular, to implement methods to reduce results variance.

Direct simulation model allows the developer to supplement the typical model logic by the special techniques that increase modeling efficiency. Of these, we consider the following:

Hierarchy of chains of events.

- Using the SELECT operator.
- Creation of specific models.
- Looped chain of events.
- Separate random number generators (RNG).
- The logarithmic scale of outputs.
- Variance reduction methods.

Chain of planned events include moments of various kinds of events: the arrival of demands (perhaps, from several independent sources); arrival of "negative" demands changing the logic of current services; service ending; exhausting application patience, etc. Note that the ordering within the chains is generally not necessary. Hierarchy of the chains of events is administered by the separately chains for each species, the choice of the nearest for each type and then comparing the lows to control the SELECT statement to assign option for nearest event processing. This further reduces the total redundancy, since each type of event is associated with processing of only a few chains. Using of SELECT greatly simplifies the program logic, eliminating complex structures with conditional statements.

An example of a specific model is the multi-channel system with interrupting priorities, which is beyond the capabilities of GPSS. Its feature is the need for a demand’s passport which contains demand type, moment of its arrival, needed service time (for interrupted ones - residual duration), moments of the service completion (for being serviced), etc. - depending on the purpose of modeling. First n positions of the array of passports present the demands in the service channels, the rest ones – in the queue. In the n-th position of the array of passports whenever you update the situation in the channels for full buzed system is convenient to put the "least respected" demand - the first candidate to interrupt. This simplifies the handling of the respective event.

Looped chains of events are convenient because when retrieving service requests do not need to shift array of passports stored in the queue: it is enough to adjust the pointers to the head and tail of the queue (when arriving application - only on the tail). Of course, this is required the overflow control using the current values of the number of entities in the system. It must be compared with a maximum length of the queue defined before the model is run.

Important condition for the correctness of the simulation is the independence of random variables of each type - the interval between applications, service time, type of applications, the successor node in the network of service, etc. The independence is achieved by using for their generation the separate RNG. For the convenience of a software implementation, it should be a kind of multiplicative generators with a common factor, but different initial values. It is required to guarantee the disjointness generated series, but constructive advices on the matter are missing in the literature. Mentioned problem can be solved by dividing the period of the generator (for a good RNG it is 2^{m-2}, where m is the number of bits) for the required number of generators and take as the initial values the corresponding numbers. To obtain these properties an algorithm can be applied which accelerates getting a random number p, based on the binary representation of its number k:

```
subroutine fastrand(k,p)
   integer k,p
   integer x0, z, j, j1 ,j2
   data x0 /57539/
   j=k; p=x0
   z=1220703125
   do while (j>0)
      j1=j/2; j2=j-2*j
      if (j2==1) p=p*z
      if (p<=0) then
         p=(p+2147483647)+1
      end if
      z=z*z
      if (z<=0) z=(z+2147483647)+1
         j=j1
   end do
end
```

In it as the current “prefabricate” p, and the degree of the factor z are dividing up on the generator module 2^{31}. Exponentiation is performed at each step, and the multiplication by p - an average of half steps, which gives the estimate of complexity the k-th number obtaining $\frac{3}{2}\log_2 k$ of multiplications. Note that the logarithm of a billion does not exceed 30. Algorithm is applicable to any initial values and multipliers and (after an obvious modification) – to any generator module. The formed integers are divided into generator module to receive the standard $\{U_i\} \in [0,1)$.

Simulation modeling is usually associated with a progressive build-up of the number of trials - preferably in a single run with the issuance of the test results in the points of logarithmic scale. Typical construction of a simulation model is the cycle WHILE which contains the choices controlling. Immediately before the end of said cycle the fragment can be inserted which multiplies after completing each order interval for m by 10:

```
if (n>n1) then
   k=mod(n,m)
   if (k==0) then
      k=n/m
   if  (k==1.or.k==2.or.k==5.or.k==10)
then
      print *, ' n = ', n
      <  output  t  of  the  desired
      results >
      n1=n
      if (k==10) m=10*m
      end if
   end if
 end if
```

Promising research in the field of simulation is the application of results variance reduction. Here we will discuss the most effective ones.

We illustrate the use of estimates with the smallest variance by two examples. With a Poissonian flow input the calculation of the stationary probabilities of the system states can be done by counting the multiplicity of states observation or through the total time spent in them (according to theorem PASTA, the received estimates must match). The latter option is preferable already by "philosophical" reasons. Another object of this kind - the calculation of moments of distribution of demands sojourn time in the network, either directly or as a convolution of the statistical moments of waiting and reliably known a priori service time distribution. In the second method one of the sources of statistical error is eliminated.

The method of antithetic variables is to use random number generators with negatively correlated results that can be obtained in different ways:

- through the "somersaulting unit" (the alternating use of *U* and 1-*U*);
- using complementary initial settings of the generators,
- by the crossing (in different runs) use of generators for intervals between arrivals and service times.

Authors did not deny the appropriateness of popular simulation environments in practical problems of particular complexity. However, in our opinion, in the fields of university education and research the using of direct modeling with the languages like modern FORTRAN is preferable. The techniques described above will help make it easier and more efficient.

3. CONCLUSION

Authors did not deny the appropriateness of popular simulation environments in practical problems of particular complexity. However, in our opinion, in

the fields of university education and research the using of direct modeling with the languages like modern FORTRAN is preferable. The techniques described above will help make it easier and more efficient.

ACKNOWLEDGEMENT
The research described in this paper is partially supported by the Russian Science Foundation (grants 14-11-0048, 14-21-00135), the Russian Foundation for Basic Research (grants 13-07-00279, 13-08-01250, 13-07-12120, 13-07-00279, 12-06-00276, 12-07-00302, 13-08-00702, grant 074-U01 is supported by Government of Russian Federation), project "5-100-2020" (arrangement 6.1.1 is supported by NRU St. Petersburg SPU), Department of nanotechnologies and information technologies of the RAS (project 2.11), by ESTLATRUS projects 1.2./ELRI-121/2011/13 «Baltic ICT Platform».

REFERENCES

Bratley P., Fox B.L. *A Guide to Simulation*. – N.Y., Springer Verlag, 1983. – 383 pp.

Law A.M., Kelton W.D. *Simulation Modeling and Analysis*. – Boston etc., McGraw-Hill, 2000. – 760 pp.

Ryzhikov Yu.I. *Simulation Modeling: Theory and Practice* (in Russian). – St. Petersburg: KORONA print,

AUTHORS BIOGRAPHY

Yuri I. Ryzhikov is the leading scientist at Russian Academy of Sciences, the Institute for Informatics and Automation, and Professor at the Military Cosmic Academy under A.F. Mojaisky, St. Petersburg. Its research interests are inventory control, numerical methods of the queuing theory and methods of direct simulation. He is the author of more than 270 scientific works.

Viacheslav A. Zelentsov is the leading researcher in Laboratory for Information Technologies in Systems Analysis and Modeling, Head of Research Consulting Center for Space Information Technologies and Systems at St. Petersburg Institute of Informatics and Automation of the RAS (SPIIRAS), professor and Director of Research and Education Center "AeroSpaceInfo" in St Petersburg State University of Aerospace Instrumentation. Fields of research: basic and applied research in integrated modelling, multi-criteria optimization under uncertainties, reliability theory, and mathematical models and methods of decision-making support in complex technical-organizational systems with the use of aerospace information.

Semyon A. Potryasaev graduated from the Baltic State Technical University "VOENMEH" with a degree of control systems engineer and Moscow Institute of International Economic Relations as an economist in finance and credit. Successfully defended his PhD thesis in 2004 at Saint Petersburg Institute of Informatics and Automation of the Russian Academy of Science (SPIIRAS). Currently works as a senior Researcher at Saint Petersburg Institute of Informatics and Automation of the Russian Academy of Science (SPIIRAS). Research interests: applied research in mathematical modeling, optimal control theory, mathematical models and methods of support and decision making in complex organization-technical systems under uncertainties and multicriteria. E-mail: spotryasaev@gmail.com.