UNE IMP

EHE ★QMIPS★ CWI

INR UER

ZAR UTO

Q uantitative
M odeling
I n
P arallel
S ystems

A solver tool for G-networks

Sophie Chabridon, Ali Labed
UFR de Math-Info
Université René Descartes
45 rue des Saints-Pères
75006 Paris, France

Marisela Hernández
LAMIFA
Université d'Amiens
33, Rue St.Leu
80039 Amiens, France

Erol Gelenbe
Dept. of Electrical Eng.
Duke University
Durham, N.C.27708-0291
USA

D W2.T1.7.v1

19 September 1995

Distribution Level: All Partners

Approved by: E. Gelenbe

Abstract In this paper, we present a graphical tool for solving a G-network model in steady state, i.e. for finding the steady-state probabilities of the number of positive customers in a G-network. The user will draw a G-network on the screen and input the parameters of each queue (mean service time, arrival rates, routing probabilities, etc.). Then the solver will provide the user with the solution of the system of non-linear traffic equations, and the stationary distribution of queue length, and performance measures such as sojourn times, mean number of customers in the system.

The QMIPS Project: The QMIPS project is a collaborative research project supported by the CEC as ESPRIT-BRA project no 7269. It is being carried out by the following organisations: CWI (Amsterdam), EHEI (University of Paris V), Imperial College (London), INRIA (Sophia-Antipolis), University of Erlangen, University of Newcastle, University of Torino and University of Zaragoza.

1 Introduction

A new class of queueing networks has been recently introduced by Gelenbe [13, 29, 30] which unify different stochastic networks such as queueing networks, computer network models and neural nets [7, 8, 9, 14, 28]. G-networks have two types of customers: positive and negative. Positive customers have the same behavior as ordinary queueing network customers. If a positive customer joins a queue it waits until it receives service or it can be destroyed by a negative customer arriving to the queue. A negative customer joining a *non-empty* queue moves or destroys a positive customer, or it will vanish immediately if the queue is empty. Negative customers do not receive service and their actions are supposed to be taken instantaneously, that is, they do not consume any time. A positive customer leaving a queue after service can join another queue remaining positive or it can become a negative customer according to some probability.

Although these networks have product form, their solution is different from that of usual BCMP models [1]. Furthermore, they also allow more complex *customers* which can destroy batches of customers or reroute traffic [30].

This paper presents a software tool which allows the user to draw a G-network on the screen and to input the parameters of each queue (mean service time, arrival rates, routing probabilities, etc.). Then the solver will provide the user with the solution of the system of non-linear traffic equations, and the stationary distribution of queue length, and performance measures such as sojourn times, mean number of customers in the system, etc.

Even though G-networks have appeared recently, they have already motivated a very important amount of research. Product form solutions have been characterized in [17, 19, 20, 33, 36, 45]. In [29], an extension of the previous work is presented where a different behavior of negative customers is considered. That is, the effect of a negative customer, now considered as a signal, arriving to a non-empty queue is to move a positive customer to another queue instead of just destroying it. This is called triggered customer movement. G-networks with signals also have product form solution [29]. In [30] the model is extended to networks where negative customers destroy batches of positive customers.

Stability conditions are provided for G-networks in [10]. Gelenbe, Glynn and Sigman [15] have studied several single server policies related to the arrival of negative customers to a non-empty queue, like removal of the customer in service and removal of the customer in the tail.

Several extensions to this model have then been considered. In [21, 22], an extension of G-networks has been discussed where positive customers could have different classes. Henderson et al. have proposed several extensions allowing state-dependent rate and batch transitions [17, 34, 37, 38]. [43] considers multiple class G-networks with jumps back to zero and [44] proposes G-networks with triggered batch state-dependent movement.

The G-network model was initially motivated by the analogy with neural networks [9]. The theory of the random neural network is developed by Gelenbe [8, 9, 23, 31]. G-networks have been widely used as neural networks, in particular with applications to the traveling salesman problem [32], minimum graph covering [24], graph partitioning [18], load balancing [41] and task assignment in a distributed system subject to failures [39]. Other applications concern problems of supervised learning of images [26], image and pattern recognition [35], texture generation [11, 27] and associative memory [16]. G-networks have been applied to evaluate the performance of unreliable flow systems [40] and to model virus behaviour in a computer network [46]. In [42], it has been proved that the solution of a G-network provides a local minimum to a quadratic cost function.

Sections 2 to 5 are devoted to the G-networks formalism for the cases mentioned, i.e. positive and negative customers, signals, batch removals. Section 6 describes the solver as a tool, that is, the user interface. In section 7 we present the numerical solution algorithm which is used and discuss some numerical aspects involved on it. In section 8 two examples concerning system reliability illustrate the use of our solver.

2 G-networks with positive and negative customers

In the simplest G-network model, customers are either *negative* or *positive.* Positive customers behave like ordinary queueing network customers. If a positive customer joins a queue it could wait until it receives service or it could be destroyed by a negative customer arriving to the queue. An arriving negative customer joining the queue instantaneously destroys a positive customer, if there is any in the queue, or it would vanish immediately if the queue is empty. Negative customers do not receive service and their actions are supposed to be taken instantaneously, that is, they do not consume any time.

External positive or negative customer arrivals to queue i constitute independent Poisson processes with rate Λ_i for positive customers and rate λ_i for negative customers. Positive customers have iid exponential service distribution times with rate r_i at queue i.

Positive customers leaving a queue after the completion of their service may join another queue either as a negative or as a positive customer. That is, a positive customer could become negative after its service is completed. The movement of customers between queues is represented by a Markov chain. A positive customer leaving queue j (after finishing service) joins queue i as a positive customer with probability $P^+_{j,i}$, or as a negative customer with probability $P^-_{j,i}$. It may leave the network with probability d_i. Let $P_{i,j} = P^+_{i,j} + P^-_{i,j}$; it represents the transition probability of a Markov chain modeling the movement of customers between queues. Therefore, we have for n such queues the following relation:

$$\Sigma_{j=1}^n P^+_{i,j} + \Sigma_{j=1}^n P^-_{i,j} + d_i = 1 \qquad 1 \leq i \leq n$$

G-networks have a non standard product form solution for the distribution of the queue length ($P(\vec{x})$) which is given in the following theorem.

Theorem 1 *[8, 13] Consider an arbitrary G-network with N queues or nodes. If the system of non-linear equations*

$$q_i = \frac{\Lambda_i + \sum_{j=1}^n r_j q_j P^+_{j,i}}{r_i + \lambda_i + \sum_{j=1}^n r_j q_j P^-_{j,i}} \tag{1}$$

has a positive solution, such that for any queue i, $q_i < 1$ holds, then the stationary distribution $P(\vec{x})$ for the queue length at each station i of the network exists and has the following product form solution:

$$P(\vec{x}) = \prod_{i=1}^n p_i(x_i = k_i)$$

where

$$p_i(x_i = k_i) = (1 - q_i)(q_i)^{k_i} \tag{2}$$

This model has been extended in [21, 22], to networks with multiple class of positive customers and just one class of negative customers and three types of service policies (Processor Sharing, FIFO, LIFO).

The only customers which sojourn in G-networks are positive ones because negative customers disappear instantaneously. So the measures considered (sojourn time, number of customers, etc.) correspond to positive customers. Positive customer departures take place when:

- a service is completed,
- a negative customer arrives to a non empty line and destroys a positive customer.

Thus the mean sojourn time W of positive customers in the system is given by:

$$W = \frac{1}{\Lambda} N$$

where Λ corresponds to the global arrival rate of positive customers to the system and is equal to $\Sigma_{i=1}^{n} \Lambda_i$.

The total number N of positive customers in the system is the sum of the mean numbers of customers at each queue i obtained from the marginal distribution:

$$N = \Sigma_{i=1}^{n} N_i = \Sigma_{i=1}^{n} \frac{q_i}{1 - q_i}$$

and q_i is given by (1).

3 G-networks with signals

In [29] an extension of the previous model is shown to have a product form solution. An arriving negative customer, which is now denoted a signal, may destroy or it may move a positive customer to another queue instead of destroying it. These movements are Markovian (the transition matrix is denoted as $Q = (Q_{i,j})$).

G-networks with signals generalize original G-networks with positive and negative customers. The following theorem for the existence of a solution of the distribution is proved in [29].

Theorem 2 *Consider an arbitrary G-network with signals. If the following system of non-linear equations*

$$q_i = \frac{\Lambda_i + \sum_{j=1}^{n} r_j q_j P_{j,i}^{+} + \sum_{m=1}^{n} (\lambda_m + \sum_{j=1}^{n} r_j q_j P_{j,m}^{-}) Q_{m,i} q_m}{r_i + \lambda_i + \sum_{j=1}^{n} r_j q_j P_{j,i}^{-}} \tag{3}$$

has a positive solution, such that for any queue i, $q_i < 1$ holds, then the stationary distribution $P(\vec{x})$ of the network exists.

With respect to performance measures such as mean sojourn times and mean number of positive customers, results which can be obtained for a queue with positive and negative customers can also be obtained for a queue in a G-network with signals.

4 G-networks with signals and batch removals

The preceding work has been generalized in [30] to allow the destruction of a batch of positive customers. A signal arriving to an empty queue will have no effect and will just disappear; if queue i is non-empty, then one of the following two events occur:

- The arriving signal triggers the instantaneous passage of a customer from queue i to some other queue j with probability $Q_{i,j}$.
- With probability $D(i) = 1 - \Sigma_j Q_{i,j}$, it forces a batch of customers of random size to leave the network. Let the length of queue i be k_i at the instant of arrival of the trigger; if $k_i \geq B_i$, then its length is reduced by B_i (batch size) and if $k_i < B_i$, the queue length becomes zero. The batch removal size distribution at queue i is given by: $P[B_i = m] = \pi_{im}, m \geq 1$.

Thus, a signal acts as an external trigger that instantaneously moves a customer from one queue to another or a batch to the outside world.

Theorem 3 *We define the function $f_i(x) = (1 - \Sigma_{m=1}^{\infty} \pi_{i,m} x^m)/(1-x)$. Then the following nonlinear equations system is used for computing the stationary distribution of the network if the following system of non-linear equations has a solution:*

$$q_i = \frac{\Lambda_i + \sum_{j=1}^{n} r_j q_j P_{j,i}^{+} + \sum_{m=1}^{n} (\lambda_m + \sum_{j=1}^{n} r_j q_j P_{j,m}^{-}) Q_{m,i} q_m}{r_i + (\lambda_i + \sum_{j=1}^{n} r_j q_j P_{j,i}^{-}) D_i f_i(q_i)} \tag{4}$$

where $D_i = 1 - \Sigma_{j=1}^{n} Qi, j$ represents the probability that a batch of random size B_i leaves the network.

Clearly, if the batch is of size 1 and the matrix Q is a null matrix, we obtain the simplest G-network with positive and negative customers described in Section 2.

5 The solver tool

This section is devoted to the presentation of the user interface of the solver. To our knowledge, this is the first tool developed for solving queueing network models which use positive and negative customers, although there exist already various tools for solving standard queueing networks: QNAP2 [2, 5], PAWS [4], RESQ [3], SIMAN V [6] etc.

The tool we propose, is written in the C language and runs on any Unix workstation provided with X-windows (version X11R5 or later). The tool allows the user to draw on the screen the G-network that is to be solved. Once it is drawn, parameters must be entered for each arc and each queue. After the solver is installed, it can be called with the command:

> *solver*

No arguments are necessary. A single window (see Figure 1) will appear on the screen. This window has three areas:

- Main Menu (at the top)

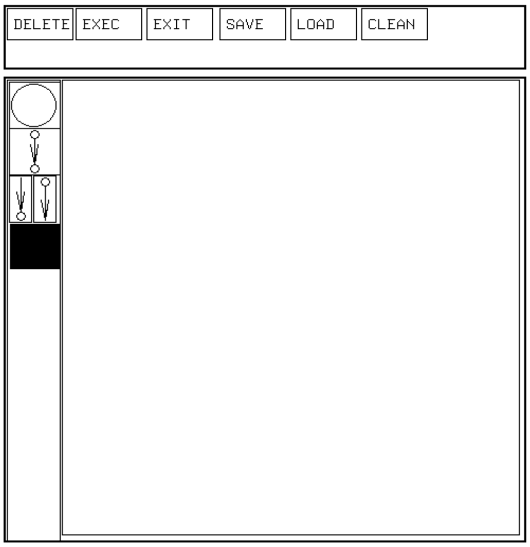

Figure 1: The solver Window

- Icons (on the left)
- The window itself which is a drawing area

We now describe how to use each of these areas. Clicking a mouse button simply consists in pressing the button and immediately releasing it. To drag with a mouse button, move the cursor over an object, then press a mouse button and hold it down. Move the mouse by continuing to hold down the mouse button and complete the drag by releasing the mouse button.

5.1 Main menu

It is localized at the top of the window. It has the following options which are selected by clicking the left button of the mouse.

There are seven possible operations. Six operations only are displayed in the main menu, and the option CREATION is the default and corresponds the case where no operation is actually in inverse video.

- CREATION : This option allows the user to create a new G-network. This is the default option of the tool.
- LOAD : It is used to load on the screen a G-network which has been previously created. A small window appears on the screen to allow the user to give the name of the file to be loaded.
- SAVE : Once created, the G-network and its corresponding parameters are saved in a file. A dialog box is displayed to ask for a file name.
- DELETE : In the process of the construction of a G-network, the user may want to delete an object. Deleting a queue will automatically delete the arrows connecting it with other queues.
- CLEAN : This option allows the user to clear what has already been drawn and to start the drawing of a new G-network.
- EXEC : This option will run the program to solve the network. The corresponding results or errors are reported in a file called “results”.

- EXIT : To exit the solver environment.

5.2 Icons

This area is placed vertically on the left of the window; it indicates the current selected icon.

There are four icons :

- node : corresponds to a service station and its associated queue; it is represented by a circle.
- connect line : corresponds to a link between two nodes; it is represented by an arrow with one circle at both end points.
- entry line : corresponds to an external arrival to a node; it is represented by an arrow with a circle at its end point.
- out line : corresponds to an external departure from a node; it is represented by an arrow with a circle at its beginning point.

5.3 Drawing area

In this area, the result of any mouse operations is function of the currently selected operation in the main menu.

- CREATE When creation mode is active, the user simply has to select an icon with the left mouse button and then click it in the drawing area at the position where the user wants the object to be drawn.

  To draw an arc from one node to another, drag from one node to another with the left mouse button.

- DELETE When the DELETE operation is selected in the main menu, clicking with the left mouse button on an object in the drawing area will automatically delete it.

When the middle mouse button is clicked on an object, a dialog box is displayed; it contains edit fields for the parameters corresponding to the clicked object type :

- Parameters of node i:
  Mu : Service rate of node i (r_i in the formulas).

- Parameters of arcs reaching node i:
  LAMBDA : Arrival rate of positive external customers (Λ_i in the formulas),
  lambda : Arrival rate of negative external customers (λ_i in the formulas).

- Parameters of arcs leaving node i:
  d : Probability for a customer to leave node i ($d(i)$ in the formulas).

- Parameters of an arc connecting nodes i and j:
  P^+ : corresponds to $P^+_{i,j}$ in the formulas,
  P^- : corresponds to $P^-_{i,j}$ in the formulas,
  Q : corresponds to $Q_{i,j}$ in the formulas.

Since this is the 1st version of this tool, there are still some features which are not yet implemented. Batch removals can not be specified; all the destructions are equivalent to the destruction of a batch of size 1. The size of the G-network is limited by the size of the window screen since there is no scrolling implemented; so the G-network can contain as many queues as can been drawn on one screen. Arrows drawing is not very flexible; an arrow is formed by only one straight line. However, when there are two arrows between two queues in opposite direction, the second arrow is automatically represented by two lines.

6 Algorithm of the solver

In [30, 29, 9] the existence of the solution of the nonlinear customer flow equations was established using Brouwer's fixed-point theorem. This is valid for stable and unstable systems. Another heuristic method for finding a solution for G-networks has been proposed in [12].

The following algorithm is proposed to compute a solution for the system of equations corresponding to the G-network under study.

1. For $i = 1, ..., n$, initialize q_i with random values in the interval $[0, 1]$ using a uniform distribution. Initialize the allowed error ε which is the maximum admitted for the square of the difference of two successive computations of q_i.
2. Compute the new value of q_i using the appropriate equation ((1), (3) or (5)).
3. Compare the value of q_i at iteration k, q_i^k, to its value at the previous iteration q_i^{k-1}.
4. Go back to step 2 if $E_1^k = \sum_{i=1}^{n}(q_i^k - q_i^{k-1})^2 > \varepsilon$.

When a solution $\widehat{q_i}$ has been found after K iterations, it can be interesting to compute at each iteration:

$$E_2^k = \sum_{i=1}^{n}(q_i^k - \widehat{q_i})^2$$

E_2^k gives the difference at the kth iteration between the current value of q_i^k and the solution value $\widehat{q_i}$, obtained at the end of the algorithm for each queue. In section 8, we present two examples of G-networks and for each of them we give the value of both E_1^k and E_2^k at each iteration.

Reaching a fixed point by means of the iterative procedure corresponds to the stationary solution of the network ($q_i < 1$ for all i); otherwise the network is not stable and there is no stationary solution.

7 Examples

In order to illustrate the use of the tool we have developed, this section is devoted to its use and application. Specifically, we first consider the model of a manufacturing system with quality control. Then we model a G-network with fifteen queues.

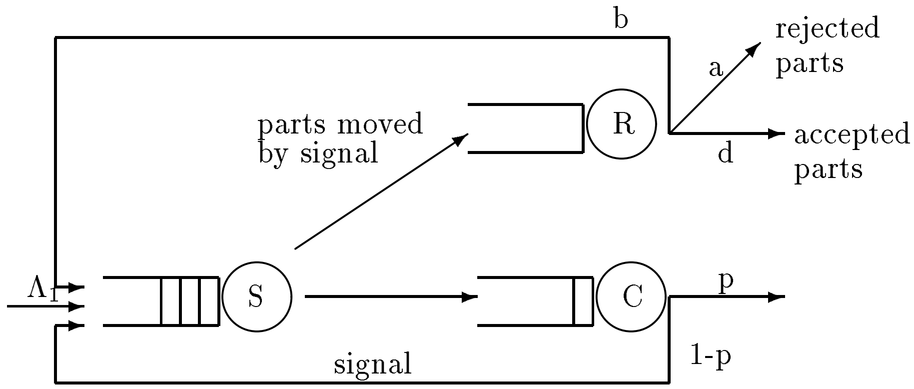

Figure 2: System 1 - A manufacturing system with quality control

7.1 A manufacturing system with quality control

On Figure 2 we show an example of a system which processes parts (system 1), originally studied in [40]. It is formed by a processing machine S and two control machines C and R. Parts arrive to the system according to a Poisson process of rate Λ_1 and are processed by machine S. Then, they are subject to inspection by machine C. At this stage, parts can be accepted with probability p, or rejected. A failing inspection in C indicates that the part that is currently being processed in station S, if any, should leave S and go through a different control process in station R. After inspection in R, parts are either accepted (with probability d), reworked at machine S (with probability b) or they leave the system. This system is modeled with a G-network with signals. The detection of a defective part in machine C is modeled by a positive customer that becomes a signal once it leaves C. Then, as a signal, it joins station S and reroutes the part in service to machine R.

On Figure 3, we give the G-Network corresponding to system 1 depicted in Figure 2. In this example, 20% of parts are rejected in C, then $1-p=0.2$. The input parameters are the following:

$$\vec{\Lambda} = \begin{pmatrix} \Lambda_1 \\ 0.0 \\ 0.0 \end{pmatrix} \qquad \vec{\lambda} = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} \qquad \vec{r} = \begin{pmatrix} 1.0 \\ 10 \\ 10 \end{pmatrix} \qquad \vec{d} = \begin{pmatrix} 0.0 \\ 0.6 \\ 0.8 \end{pmatrix}$$

$$P^+ = \begin{pmatrix} 0.0 & 0.0 & 1.0 \\ 0.4 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \qquad P^- = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.0 \end{pmatrix} \qquad Q = \begin{pmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}$$

The corresponding results obtained using the solver are given below:

$$\vec{q} = \begin{pmatrix} 8.590353E-01 \\ 1.475883E-02 \\ 8.590353E-02 \end{pmatrix}$$

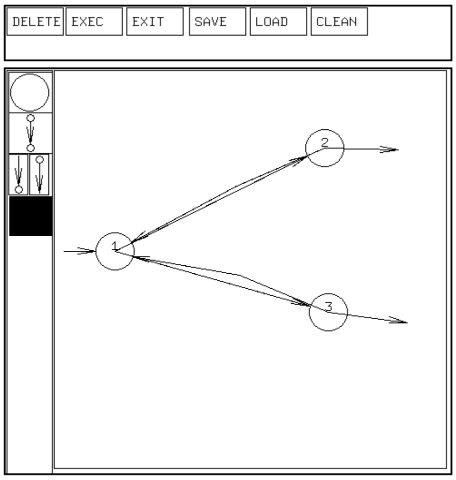

Figure 3: G-network corresponding to system 1

Average number of customers in queue S : 6.093976E+00
Average number of customers in queue R : 1.497992E-02
Average number of customers in queue C : 9.397644E-02

Average of total number of customers in the system: 6.202933E+00
Average sojourn time in the system : 7.753666E+00

Varying Λ_1, the arrival rate of parts to machine S, we obtain the curve shown in Figure 4 for the average of the total number of parts in the system. Figure 5 gives the number of iterations that was necessary for the solver to reach a fixed point for each value of Λ_1. On Figure 6, we give the value of the errors E_1 and E_2 at each iteration when Λ_1 is equal to 0.8; in this case there were 29 iterations.

7.2 A G-network with 15 queues

As a second example of the utilization of the solver, we present in Figure 7 a G-network of fifteen queues. For this example, each queue has only two successor queues (except for queues 13, 14 and 15). We have supposed that all queues have the same service rate (r).
The input parameters of the G-network of Figure 7 are listed below, where $\frac{p}{2}$ is the probability that a positive customer will become negative after leaving a queue (see matrix P^-).

$$\vec{\Lambda} = (0.5, 0.3, 0.4, 0.2, 0.7, 0.6, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$\vec{\lambda} = (\frac{p}{2}, \frac{p}{2}, \frac{p}{4}, p, \frac{p}{3}, p, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$\vec{d} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1)$$

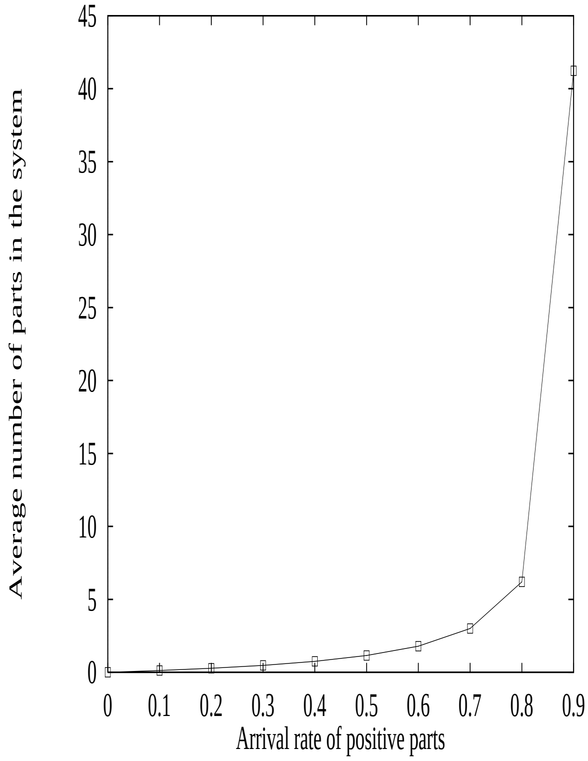

Figure 4: Average number of parts in the system versus the arrival rate Λ_1

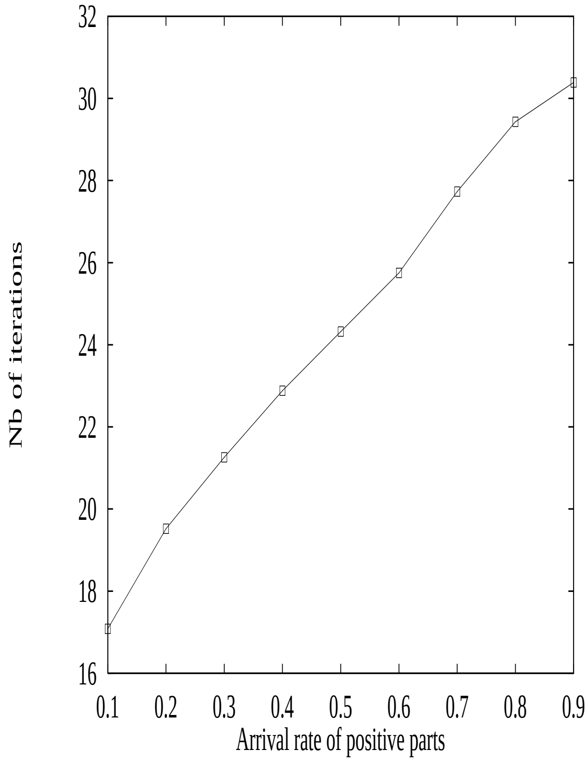

Figure 5: Number of iterations versus the arrival rate Λ_1

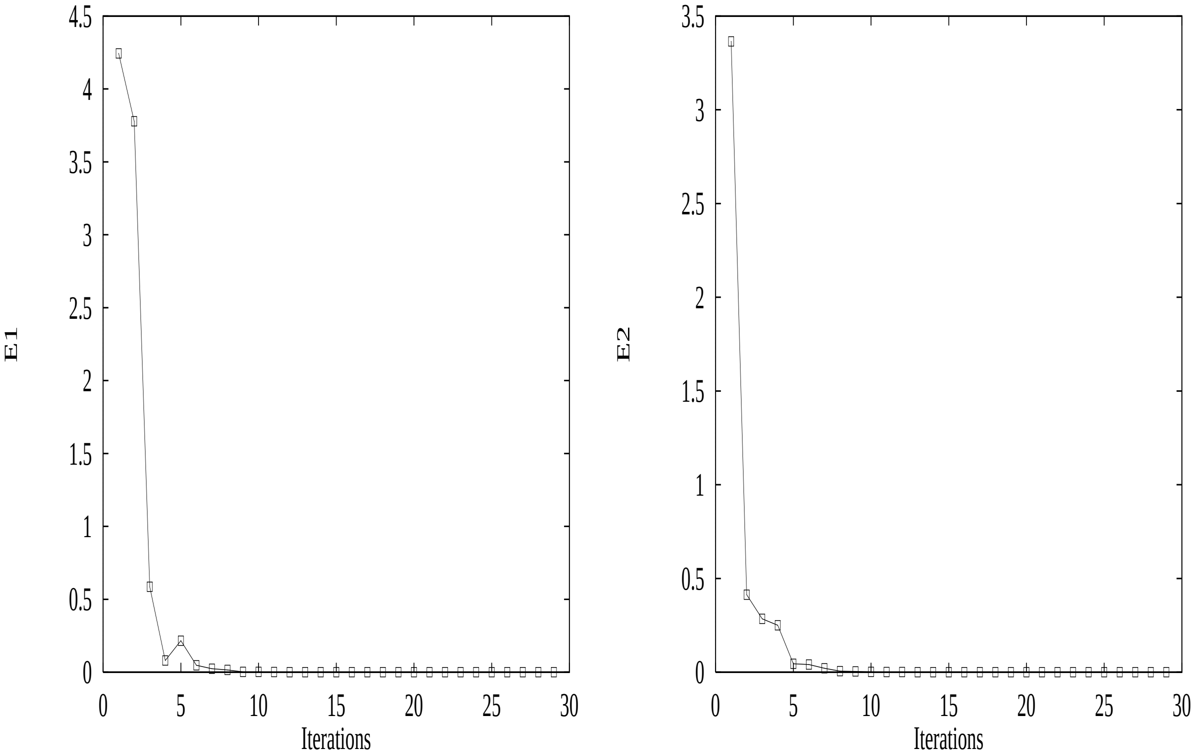

Figure 6: Vari = 0.8

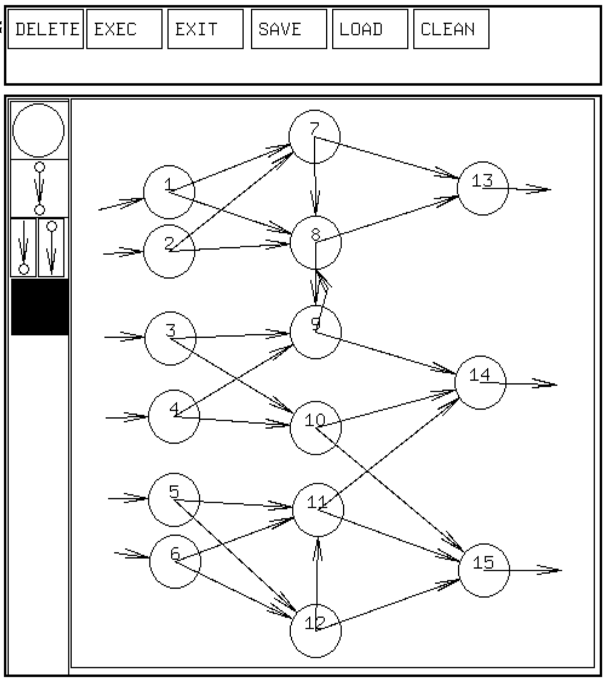

Figure 7: A G-network with 15 queues

$$
P^{+} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & 0 & 0 & 0 & 0 & \frac{1-p}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & 0 & 0 & 0 & \frac{1-p}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & \frac{1-p}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p}{2} & 0 & 0 & 0 & \frac{1-p}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

$$
P^{-} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & 0 & 0 & 0 & 0 & \frac{p}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & 0 & 0 & 0 & \frac{p}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & 0 & 0 & 0 & \frac{p}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

$$
Q = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p}{2} & \frac{p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

The analytical solution of this system is very cumbersome; it requires to solve a system of 15 non-linear equations with 15 unknowns. With the iterative method used by the solver, the solution was found in no more than 29 iterations.

Varying the parameter p (probability of becoming negative), we obtain the curves shown in Figure 8 for three different service rates ($r = 1, 5$ and 10). They give the average of the total number of positive customers present in the system in the stationary state and also their average sojourn time. We easily see that when p increases the number of positive customers in the system decreases, i.e. as p increases, there are more positive customers becoming negative and consequently destroying positive customers. Therefore, N, the total number of positive customers decreases. In Figure 9, we give the number of iterations performed by the solver before reaching a fixed point solution for values of p in [0.1,1]. On Figure 10, we give the value of the errors E_1 and E_2 at each iteration when p is equal to 0.5; in this case 14 iterations were necessary to reach the solution.

In the following, we compare the behaviour of the network from Figure 7 without signals with the same network in the presence of signals. Notice that signals are present only in the first layer of the network. That is from queues 1 to 7, 1 to 8, 2 to 7, 2 to 8, 3 to 9, 3 to 10, 4 to 9, 4 to 10, 5 to 11, 5 to 12, 6 to 11 and 6 to 12. Figure 11 gives the number of positive customers for $r = 1$ with respect to p with and without signals. We can see that when there are signals, there are more positive customers in the system than without signals because in this case they are rerouted in the first layer instead of being destroyed.

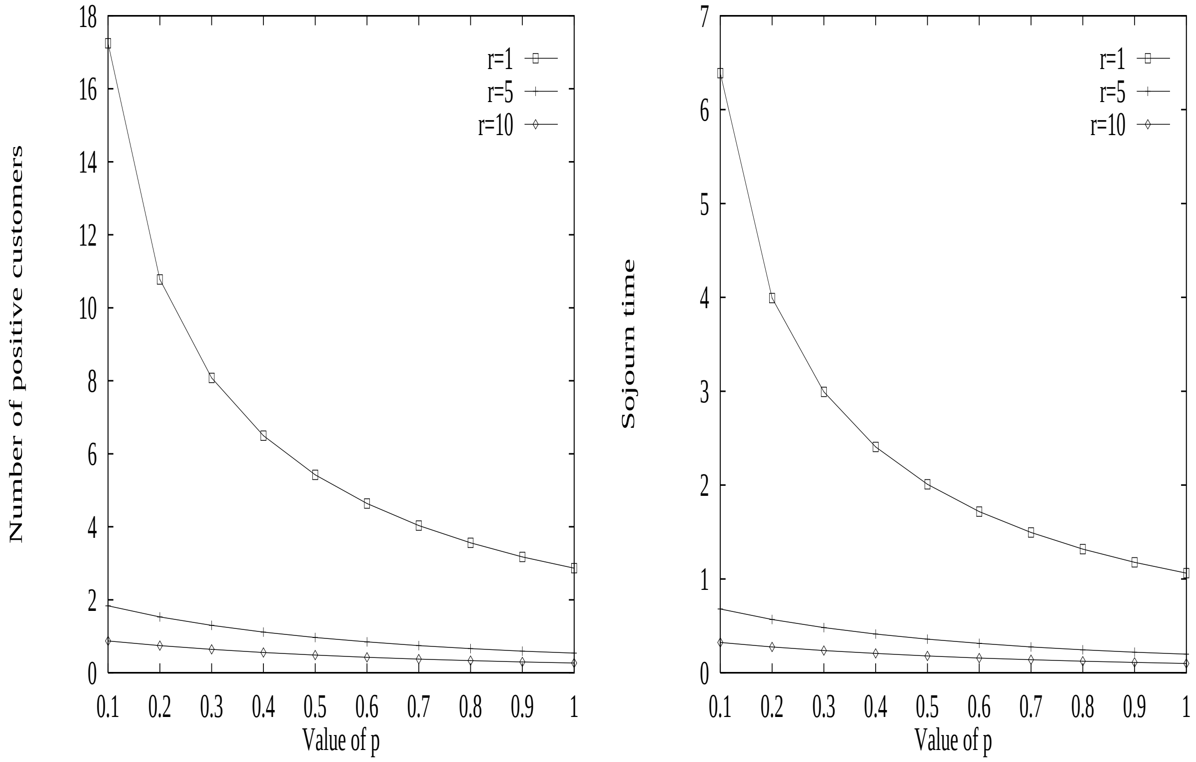

Figure 8: Average total number of positive customers and their sojourn time in the system versus p

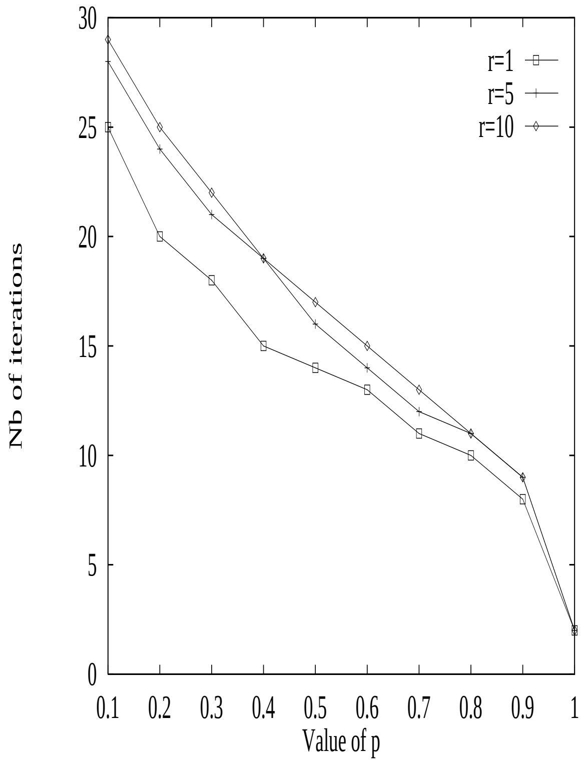

Figure 9: Number of iterations versus p

8 Conclusions

The purpose of this paper has been to present a tool for solving the G-network model in steady state. Due to the fact that this formalism is being widely used, a tool such like this facilitates the task of modeling practical applications. Fast results can be found using the solver for large G-networks. Model parameters as well as the results are given in a user-friendly way. Further work will allow the solver to automatically vary certain parameters presenting curves of the performance measures required.

References

[1] Baskett F., Chandy K.M., R.R. Muntz, Palacios F.G., *Open, Closed and Mixed Networks of Queues with Different Classes of Customers*, Journal of ACM, 22(2):248-260 (Apr. 1975).

[2] Merle D., Potier D., Véran M., *A Tool for Computer Systems Performance Analysis*, Performance of Computer Installations, Ed. Ferrari D., Amsterdam, North-Holland, pp. 195-213 (1978).

[3] Sauer C.H., MacNair E.A., *Simulation of Computer Communication Systems*, Englewood Cliffs, Prentice-Hall, USA (1983).

[4] *PAWS: A User Guide*, Information Research Associates, Austin, TX, USA (1983).

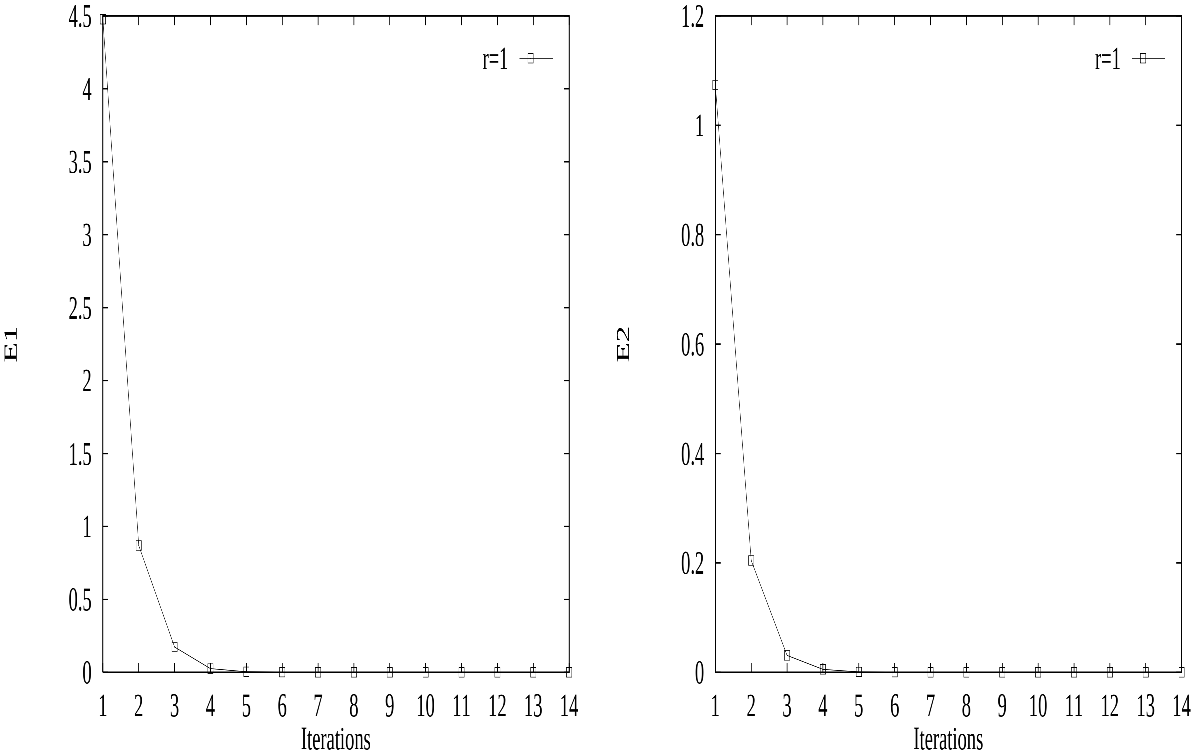

Figure 10: Variation of E_1 and E_2 during the iterations for $p = 0.5$

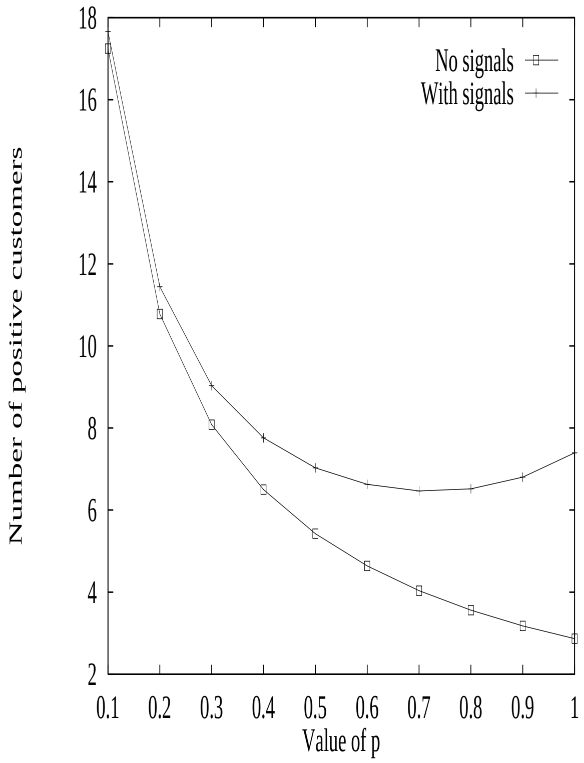

Figure 11: Number of positive customers with and without signals

[5] Véran M., Potier D., *QNAP2: A Portable Environment for Queueing Systems Modeling*, Int. Conf. on Modeling Techniques and Tools for Performance Analysis, North-Holland, pp. 25-63 (1984).

[6] Pegden C.D., *Introduction to Siman*, Systems Modeling Corp., State College, PA, USA (1984).

[7] Gelenbe E., *Random Neural Networks with Negative and Positive Signals and Product Form Solution*, Neural Computation, 1(4):502-510 (1989).

[8] Gelenbe E., *Réseaux Stochastiques avec Clients Négatifs et Positifs et Réseaux Neuronaux*, Comptes-Rendus Académie des Sciences, 309, Série II, Paris, France, pp.979-982 (1989).

[9] Gelenbe E., *Réseaux Neuronaux Aléatoires Stables*, Comptes-Rendus Académie des Sciences, Paris, France, pp.310-313 (1990).

[10] Gelenbe E., *Stability of the Random Neural Network Model*, Neural Computation, 2(2):239-247, (1990).

[11] Atalay V., Gelenbe E., Yalabik N., *Image Texture Generation with the Random Neural Network Model*, Int. Conf. on Artificial Neural Networks (ICANN-91), Helsinki (Kohonen T. ed.), Elsevier, (1991).

[12] Fourneau J.M., *Computing the Steady-state Distribution of Networks with Positive and Negative Customers*, LRI Report, $13^{th} XS$ IMACS World Congress on Computation and Applied Mathematics, Dublin, Ireland (1991).

[13] Gelenbe E., *Product Form Queueing Networks with Positive and negative customers*, Journal of Applied Probability, 28:656-663 (1991).

[14] Gelenbe E., *Theory of the Random Neural Network Model*, Neural Networks: Advances and Applications (E. Gelenbe ed.), North-Holland (1991).

[15] Gelenbe E., Glynn P., Sigman K., *Queues with Negative Arrivals*, Journal of Applied Probability, Vol. 28, pp. 245-250 (1991).

[16] Gelenbe E., Stafylopatis A., Likas A., *Associative Memory Operation of the Random Network Model*, Proc. Int. Conf. on Artificial Neural Networks, Amsterdam, North-Holland (1991).

[17] Henderson W., Northcote B.S., Taylor P.G., *Modeling using queueing networks with signals*, ITC 13 (1991).

[18] Aguilar J., *Comparison between the Random Neural Network Model and other Optimization Combinatorial Methods for the Large Acyclic Graph Partitioning Problem*, Proc. 7th Int. Symp. on Computer and Information Sciences (ISCIS), Antalya, Turkey (1992).

[19] Boucherie R., *Product-form in queueing networks*, PhD thesis, Vrije Universiteit, North-Holland (May 1992).

[20] Boucherie R., Van Dijk N., *Local Balance in queueing networks with negative customers*, Research memorandum 1992-1, Free University of Amsterdam, North-Holland (1992).

[21] Fourneau J-M., Gelenbe E., *G-Networks with Multiple Classes of Signals*, Proceedings ORSA Computer Science Technical Committee Conference, Williamsburgh, VA, USA, Jan. 8-10, Pergamon Press (1992).

[22] Fourneau J.M, Gelenbe E., *Multiclass G-Networks*, ORSA Conference: Computer Science and operation Research : new developments in their interface, Williamsburg, USA (Jan. 1992).

[23] Gelenbe E., *G-Nets and Learning Recurrent Random Networks*, Proc. Int. Conf. on Artificial Neural Networks, Brighton, England (1992).

[24] Gelenbe E., Batty F., *Application of the Random Neural Network Model to the Minimum Graph Covering*, Neural Networks: Advances and Applications 2 (E. Gelenbe ed.) North-Holland, (1992).

[25] Gelenbe E., Schassberger P., *Stability of Product Form G-Networks*, Probability in Engineering and Informational Sciences, Cambridge University Press, England, Vol. 6, pp. 271-276 (1992).

[26] Hubert C., *Supervised Learning and Retrieval of Simple Images with the Random Neural Network*, Proc. 7th Int. Symp. on Computer and Information Sciences (ISCIS), Antalya, Turkey (1992).

[27] Atalay V., *Réseaux de neurones aléatoires et textures d'images*, PhD Thesis, University of Paris V, France (Nov. 1993).

[28] Gelenbe E., *G-Networks: A Unifying Model for Neural Nets and Queueing Networks*, Proc. of the 1st Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'93), San Diego, CA, USA, Simulation Series, 25(1), pp. 3-8 (Jan. 1993).

[29] Gelenbe E., *G-Networks with Triggered Customer Movement*, Journal of Applied Probability, Vol. 30, pp. 742-748 (1993).

[30] Gelenbe E., *G-Networks with Signals and Batch Removals*, Probability in Engineering and Informational Sciences, Cambridge University Press, England, Vol. 7, pp. 335-342 (1993).

[31] Gelenbe E., *Learning in the Recurrent Random Neural Network*, Neural Computation, 5(5):154-164, (1993).

[32] Gelenbe E., Koubi V., Pekergin F., *Dynamical Random Neural Network Approach to the Traveling Salesman Problem*, Proc. Conf. on Systems, Man and Cybernetics, Illinois, USA (1993).

[33] Harrison P.G., Pitel E., *Sojourn Times in Single Server Queues with Negative Customers*, Journal of Applied Probability, Vol. 30, pp. 943-963 (1993).

[34] Henderson W., *Queueing networks with negative customers and negative queue lengths*, Journal of Applied Probability, Vol. 30, pp. 931-942 (1993).

[35] Mokhtari M., *Réseau Neuronal Aléatoire: Applications à l'apprentissage et à la reconnaissance d'images*, PhD Thesis, University of Paris V, France (Jan. 1994).

[36] Harrison P.G., Pitel E., *The M/G/1 Queue with Negative Customers*, Proceedings of the 4th QMIPS (Queueing Modeling in Parallel Systems) Workshop, London, England, pp. 185-213 (Apr. 1994).

[37] Henderson W., Northcote B.S., Taylor P.G., *Geometric equilibrium distributions for queues with interactive batch departures*, Annals of Operations Research, Vol. 48, pp. 493-511 (1994).

[38] Henderson W., Northcote B.S., Taylor P.G., *Networks of customer queues and resource queues*, ITC 14 (1994).

[39] Hernández M., Aguilar J., *A Simulator for Task Assignment in a Distributed System Subject to Failures*, Proc. of the 4th QMIPS (Queueing Modeling in Parallel Systems) Workshop, London, England, pp. 89-106 (Apr. 1994).

[40] Hernández M., Fourneau J-M., *Modeling Defective Parts in a Flow System using G-Networks*, Proc. Workshop on Performability Modeling of Computer and Communication Systems, Le Mont St-Michel, France (Jun. 1993).

[41] Aguilar J., *L'allocation de tâches, l'équilibrage de charge et l'optimisation combinatoire*, PhD Thesis, University of Paris V, France (1995).

[42] Gelenbe E., *G-networks and Minimum Cost Functions*, Proc. of the 3d Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'95), pp. 135-141, Durham, NC, USA (Jan. 1995).

[43] Fourneau J-M., Kloul L., Quessette F. *Multiple Class G-Networks with Jumps back to Zero*, Proc. of the 3d Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'95), Durham, NC, USA, pp. 28-32 (Jan. 1995).

[44] Fourneau J-M., Verchère D. *G-Networks with Triggered Batch State-dependent Movement*, Proc. of the 3d Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'95), Durham, NC, USA, pp. 33-37 (Jan. 1995).

[45] Harrison P.G., Pitel E., *Response Time Distributions in Tandem G-networks*, To appear in Journal of Applied Probability (Mar. 1995).

[46] Hernández M., *Virus Transmission in a Computer Network*, Research Report LAMIFA, Université d'Amiens, France, to appear (1995).