

# Random Search as a Means for Queuing Network Optimization

Ziny Flikop

UDC 519.95

Translated from *Kibernetika*, No.4 pp.75-79, July-August, 1969. Original article submitted January 3, 1968.

© 1972 Consultants Bureau, a division of Plenum Publishing Corporation, 227 West 17<sup>th</sup> Street, New York, N.Y. 10011. All rights reserved. A copy of this article is available from the publisher for \$15.00.

One of the problems in queuing theory is to find the optimal structure of a complex queuing network. The optimal network should offer minimal operational loss due to idle periods of the servers and served units. In most cases it is impossible to evaluate the operational quality of a branched multiphase network by analytical means. Hence the wide popularity of the statistical modeling method, which uses a stochastic model to determine quite accurately the network characteristics. The method also enables a simple random search algorithm to be used for optimization.

Consider a queuing network specified by a stochastic model and consisting of  $n$  servicing systems. The network state is defined by a vector

$$X = (X_1, X_2, \dots, X_i, \dots, X_n) \quad (1)$$

where  $X_i$  is the number of servers in the  $i$ -th system,  $i=1,2,\dots,n$ .

Denote by  $A$  the set of available states; for each  $X \in A$  there is a corresponding value of the target function  $Q=Q(X)$ . It will be assumed that  $Q(X)$  has a  $j$ -th local minimum at

$$X_{0j} \text{ if } Q(X_{0j}) \leq Q(X_e) \quad j=1,2,\dots \quad (2)$$

for  $\forall X_e \in A_{e_j}$ , where  $A_{e_j}$  is the neighborhood of the point  $X_{0j}$ . In view of the fact, that the  $X_i$  must be positive integers, by  $A_{e_j}$  is meant the set of points  $X \in A$  for which

$$X_i - X_{e_j i} = d : d = -1; 0; +1. \quad (3)$$

If the network is not at minimum, then  $Q(X_e) > Q(X_{0j})$  for at least one  $X_e \in A_{e_j}$ , where  $X_{0j}$  is, this case, the position of the network. If the points  $X_e \in A_{e_j}$  are selected equiprobably for checking, we can find one at which the value of the target functions less than  $Q(X_{0j})$ . After such a "successful" attempt, we translate the network to the newly found  $X_{0j+1} \in A_{e_j}$  and confine the check to its neighborhood. On finding  $X_e \in A_{e_{j+1}}$

with  $Q(X_e) \rho Q(X_{j+1})$ , we translate the network again, to  $X_{0j+2} \in A_{ej+1}$ . This process continues till all attempts to find a point in  $A_{ej}$  with  $Q(X_e) \rho Q(X_{0j})$  prove fruitless. The number of points in a neighborhood in the absence of constrains is  $3^n - 1$ , while the number of checked  $X_e \in A_{ej}$  is primarily determined by the time for evaluating  $Q(X)$ .

If a complete check is made of  $A_e$ , the exact coordinates of the local minimum will be found. If only part of the neighborhood is investigated, we can speak of the network being at a local minimum with some probability  $P$ , which depends on the number of available points in the neighborhood, the number of  $X_e \in A_{ej}$  for which  $Q(X_e) \rho Q(X_{0j})$ , and how many of  $X_e \in A_{ej}$  are checked.

An incomplete investigation of a neighborhood may be organized in two ways.

1. After each unsuccessful attempt to find in  $A_e$  a point with  $Q(X_e) \rho Q(X_{0j})$ , Bayes formula is used to evaluate  $P$  allowing each time for the reduction in the number of unchecked  $X_e \in A_{ej}$ . The attempts cease of reaching  $P \geq a$ , where  $a$  some previously assigned figure.
2. We first find the number  $N$  of points the investigation of which gives  $P \geq a$ . For this purpose we use the empirical formula

$$N = P(S+1) - 1 \quad \{ (1/(S+1)) \leq P \leq 1 \} \quad (4)$$

where  $S$  is the total number of checked points.

For sampling the queue from  $X_e$  to  $X_{0j}$ , a testing vector  $X^0 = (X_1^0, X_2^0, \dots, X_n^0)$  is added, in which  $z$  components take the values  $+1$  and  $-1$  equiprobably, while the remaining  $q = n - z$  are equal to zero, and  $P(X_1^0 = 0) = \dots = P(X_q^0 = 0)$ , where  $P(X_i^0 = 0)$  is the probability of obtaining  $X_i^0 = 0$ . Denote by  $X_{0j}^0$  the vector  $X^0$  enabling a point with  $Q(X_e) \rho Q(X_{0j})$ , to be discovered in  $A_{ej}$ .

In this type of neighborhood, the points are arranged in layers relative to  $X_{0j}$ . The distance  $R_p$  from  $X_{0j}$  to the  $p$ -th layer is given by

$$R_p = \sqrt{n - P} \quad (5)$$

The number of points in the  $p$ -th layer [1] is:

$$L_p = 2^{n-p} \frac{n!}{p!(n-p)!} \quad (6)$$

The neighborhood is best checked in layers, the number  $N_p$  of points checked in the  $p$ -th layer be dependent on  $L_p$  and  $a$ . The zero layer is inspected first. (If the test is successful, the network is displaced over the greatest distance.) If, after investigation of  $N_0$  points in the zero layer, no  $X_e$  is discovered with  $Q(X_e) \leq Q(X_{0j})$ , the first layer is examined, then if this fails, the second layer and so on, till all layers have been checked. To speed up the optimization, when transferring the network from  $X_{0j}$  to  $X_0$ , the coordinates of the next check point must be found by adding  $X_{0j+1}$  to  $X_{0j}^0$ .

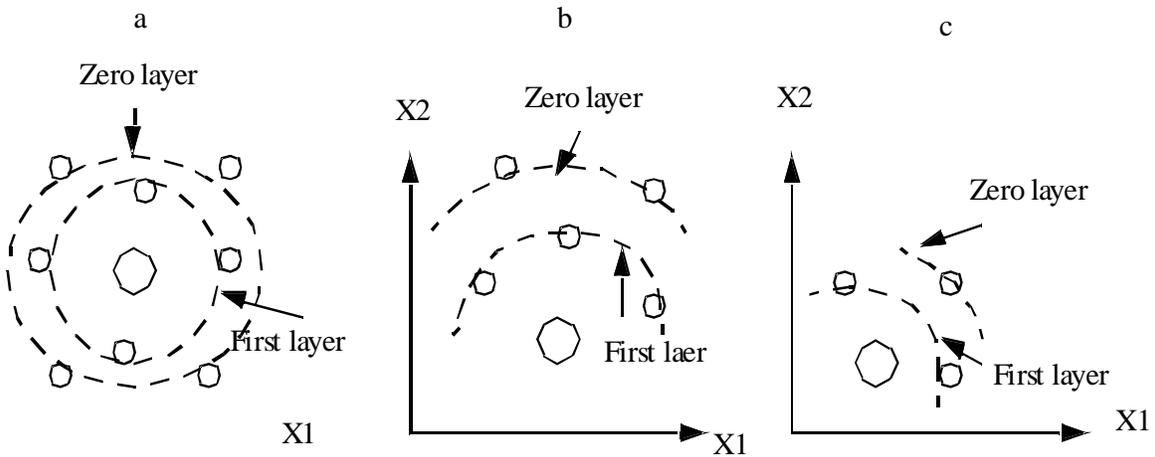


Fig. 1. Number of available points in neighborhood:

- a)  $X_{1,2} > 1$ ; b)  $X_1$  or  $X_2 = 1$ ;
- c)  $X_{1,2} = 1$ .

As an example, consider the process of finding the optimal number of servers in the stochastic model of a queuing network consisting of a source with an limited number of demands and to serving systems connected in parallel. The network operation is organized in such a way that every demand, served in the system, is a return to the source. In view of this, change in the operational mode of one system changes the operating of conditions in other (the number of demands present at the source depends on the operational quality of both systems).

The model state is defined by a vector  $X = (X_1, X_2)$  where  $X_{1,2}$  is the number of servers in the first (second) system. The network operating quality will be assessed in terms of the total losses due to idle periods of the servers and served elements. The value of this target function depends on  $X_1$  and  $X_2$ , i.e.,  $Q(X)$ .

The restriction  $X_{1,2} \neq 0$  is imposed on the vector  $X$  (severs cannot be allowed absent in the system). The number of available points in a neighborhood is thus defined by the position of the network. When  $X_{1,2} \neq 1$ , the number of  $X_e \in A_{ej}$  is a maximum (four points each in the zero and first layer). If one of the coordinates of  $X_{0j}$  is equal to unity, there will be two points in the zero layer and three in the first. When  $X_{1,2} = 1$ , the number of available points is a maximum (one in the zero and two in the first layer).

Two types of memory,  $M$  and  $M_{ej}$  are required operation the optimizing program. The memory  $M$  stores all that previously checked  $X$ , and  $M_{ej}$  only the  $X_e \in A_{ej}$ . The number of previously inspected  $X_e \in A_{ej}$  is found by means of  $M_{ej}$ , while  $M$  enables the repatriation of checked points to be avoided. It is assumed that  $X_e \in A_{ej}$ , not contained in,  $M_{ej}$  is selected for investigation. If  $X_e$  is stored in  $M$ , then  $Q(X_e) \neq Q(X_{0j})$ . If  $X_e$  is not present in  $M$  or  $M_{ej}$ ,  $Q(X_e)$  is evaluated by means of simulation.

Using Buslenko's notation [2], the operator scheme of the optimization algorithm is:

$$\Phi_1^{6,26} P_{2\downarrow 4} \Phi_3^7 \Phi_4^3 P_{5\downarrow 1} A_6 P_{7\downarrow 17} A_6 F_9 A_{16} A_{11} A_{12} F_{13} A_{14} P_{15\downarrow 23}^{24} P_{16\downarrow 25}^{\uparrow 23} A_{17}^{7,25} K_{18} P_{19}^{\uparrow 27} P_{20\downarrow 22} F_{21} F_{22}^{20} \Phi_{23}^{15,16,21} \\ A_{24}^{15} P_{25}^{16\uparrow 17} A_{26}^{1,19} L_{27}$$

Here  $\Phi_1$  is the source of demands,  $P_2$  the selection of serving system,  $\Phi_3$  the first serving system,  $\Phi_4$  this second serving system,  $P_5$  determination of the instant when transfer into the net work terminates,  $A_6$  evaluation of the target function on the basis of the queuing system operational results,  $P_7$  comparison of the target function values obtained with the least value obtained,  $A_8$  storage of the least value of the target function and the corresponding vector,  $F_9$  determination in the light of the boundary conditions of the number of available points in the neighborhood,  $A_{10}$  clearance of memory  $M_{ej}$  containing the coordinates of the points of the  $j$ -th neighborhood,  $A_{11}$  storage of the network coordinates in  $M_{ej}$ ,  $A_{12}$  clearance of the unsuccessful checks counter,  $F_{13}$  adjustments of the program for generating the test vector to obtaining  $X_{1,2}^0 \neq 0$ ,  $A_{14}$  selection of the next point to be checked in the discovered "successful" direction,  $P_{15}$  check of the bindery conditions,  $P_{16}$  checked of memory  $M_{ej}$ , so as to a void repeat. inspection of points of the  $j$ -th neighborhood,  $A_{17}$  storage of the coordinates of  $X_e \in A_{ej}$  for which  $Q(X_e) \neq Q(X_{0j})$ ,  $K_{18}$  the unsuccessful checks counter,  $P_{19}$  the determination of whether unchecked points are present in the  $j$ -th neighborhood,  $P_{20}$  determination of whether unchecked points are present in the zero layer,  $F_{21}$  adjustment of the program

for generating the tasting vector to obtaining  $X_{1,2}^0 \neq 0$ ,  $F_{22}$  adjustment of the program generating the tasting vector to obtaining one coordinate equal to zero where  $P(X_1^0 = 0) = P(X_2^0 = 0)$ ,  $\Phi_{23}$  obtains the testing vector,  $A_{24}$  determines the coordinates of the next check point,  $P_{25}$  checks the memory  $M$ , so that repeat inspection of  $X_e \in A_{ej}$  is avoided,  $A_{26}$  stores the coordinates of points checked by simulation, and  $L_{27}$  prints the minimum value of the target function and the coordinates of the corresponding point, and stops the algorithm.

The optimization process will be discussed with help of described above algorithm.

Demands appearing at the source  $\Phi_1$  are directed by operator  $P_2$  in accordance with the accepted law to one of the serving systems ( $\Phi_3$  or  $\Phi_4$ ), whence they are returned to  $\Phi_1$ . The nature of the process is estimated by  $P_5$  which realizes the transfer to  $A_6$  when the model reaches a stationary state. The operator  $A_6$  evaluates  $Q(X_e)$  from the simulation data, while  $P_7$  compares the value obtained for the target function with the previously found minimum [apart from  $Q(X_{01})$ ], which is compared with some figure known to be larger]. If  $Q(X_e) \leq Q(X_{0j})$ , we store  $X_e$  in  $M_{ej}$  and increase by one the content of the counter  $K_{18}$  ( $K_{18}$  stores the number of checked  $X_e \in A_{ej}$  for which  $Q(X_e) \leq Q(X_{0j})$ ). Next,  $P_{15}$  determines whether all the available points of the neighborhood has been inspected. If there are uninspected points in  $A_{ej}$ ,  $P_{20}$  checks the quality of investigation of the zero layer. If there are uninspected  $X_e$  in the zero layer,  $\Phi_{23}$  is adjusted by  $F_{21}$  for generating of the testing vector  $X^0$  with coordinates  $X_{1,2}^0 \neq 0$ ; otherwise it is adjusted by  $F_{22}$  to obtaining  $X^0$  with  $X_1^0 = 0$  or  $X_2^0 = 0$ . The testing vector  $\Phi_{23}$  is applied to  $A_{24}$  which evaluates  $X_e$ . If  $X_{1,2}^0 = 0$ , the point found for checking is not accepted by operator  $P_{15}$ . In this case  $\Phi_{23}$  generates  $X^0$  afresh, ...*(it is unreadable text here)* ...the check, a new point is selected by means of the circuit  $P_{15 \downarrow 23} P_{16 \downarrow 25}^{\uparrow 23} \Phi_{23}^{15,16} A_{24}^{15}$ . The resulting uninspected  $X_e$  is checked by  $P_{25}$  which, if  $X_e$  is present in  $M$ , carries out the transfer to  $A_{17}$  and so on (the point was previously checked and  $Q(X_e) \leq Q(X_{0j})$  for it), while if  $X_e$  is not in  $M$ , it transfers to  $A_{26}$  which store  $X$  in  $M$ . The value of  $Q(X)$  is found by means of the model for the new unchecked  $X$ .

Sense of the condition  $Q(X_e) \leq Q(X_{0j})$  has already been considered; let us examine the program operation with  $Q(X_e) \geq Q(X_{0j})$ . In this case  $Q(X_{0j})$  and  $X_{0j}$  are stored by the operator  $A_8$  (the network moved to a new point). Preparation is made for complete check of the neighborhood  $A_{ej}$ . For this 1)  $F_9$  determines the number  $X_e \in A_{ej}$  in accordance with the boundary conditions;  $A_{10}$  clears  $M_{ej}$ ; 3)  $A_{11}$  stores  $X_{0j}$  in  $M_{ej}$ ; 4) the operator

clears  $A_{12}$  the unsuccessful attempts counter; 5)  $F_{13}$  adjusts  $\Phi_{23}$  for obtaining  $X^0$  with  $X_{1,2}^0 \neq 0$ . Next,  $A_{14}$  evaluates the coordinates of the next point by adding  $X_{0j+1}$  to the next testing vector. The  $X_e$  obtained goes to the check circuit  $P_{15 \downarrow 23} P_{16 \downarrow 25}^{\uparrow 23}$ . If the  $X_e$  does not satisfy the conditions of operators  $P_{15}$  and  $P_{16}$ , another point is selected by means of  $\Phi_{23}$ . During the passage of  $X_e$  through  $P_{15}$  and  $P_{16}$  the operator  $P_{25}$  either regards the attempt as unsuccessful and transfers to  $A_{17}$  or after storing  $A_{26}$ , it obtains  $Q(X)$  by means of the model. The optimization process terminates when  $P_{19}$  generates a characteristic signal indicating that for all  $X_e \in A_{ej}$   $Q(X_e) \nabla Q(X_{0j})$ . After this  $L_{27}$  prints  $Q_{\min}$  and  $X_{0j}$ , and stops the process.

To reduce the search time, the initial point should be as close as possible to is the optimal; this is achieved by using experience gained with like queuing networks.

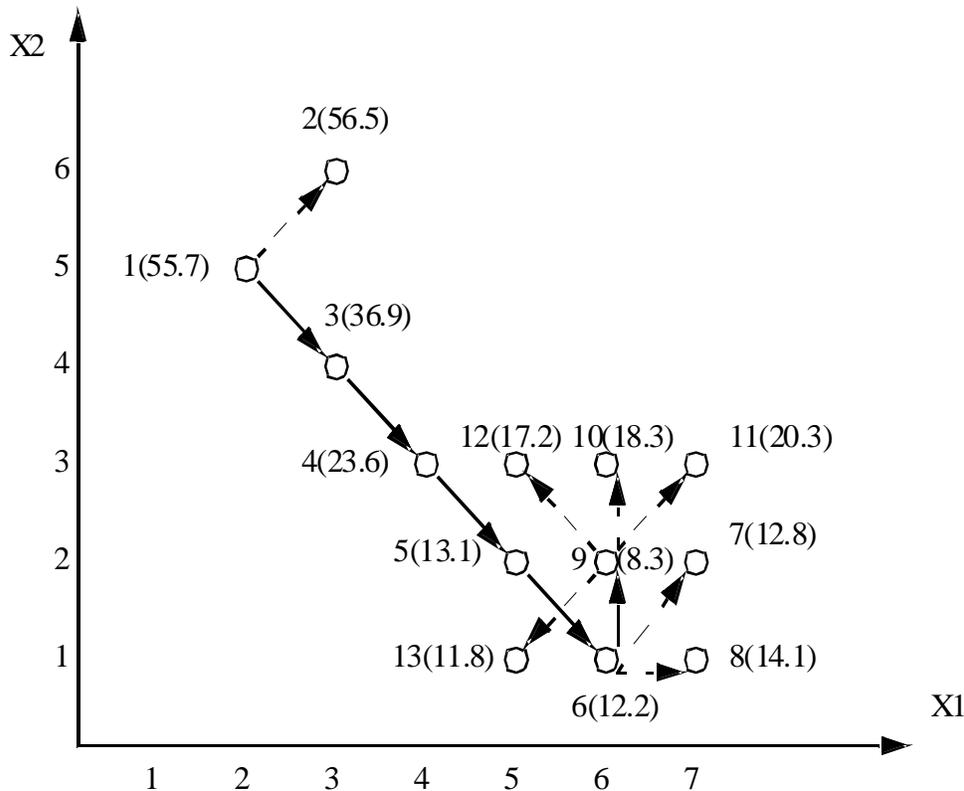


Fig. 2. Path of optimized network. Point # 1 is initial state of network; point # 9 is optimal state of network; points 2 - 8 and 10 - 13 are intermediate network states;  $\rightarrow$  is network path;  $- \rightarrow$  unsuccessful attempts. The number of attempts and corresponding value of  $Q(X)$  are represented in parentheses.

Figure 2 illustrated the optimization process for a queuing network model, consisting of two parallel systems (the algorithm of the model was developed by I.A. Luik [3]). The following network characteristics were chosen for our illustration. The input flow is Poisson with intensity  $\lambda = g\lambda'$  where  $g$  is the number of units present in the source and  $\lambda' = 0.2h^{-1}$  is the intensity of input demands from one unit. The serving time is exponential. The serving intensity by one server of the first system is  $\mu = 1h^{-1}$  and the second system,  $\mu = 3h^{-1}$ . The cost of idling time per unit is 0.5 rubles/h per server of the first system 2 rubles/h, and per server of the second system 10 rubles/h. The total number of units  $g=50$ , demand is directed equiprobably to either of the serving system.

The initial state of the network was assumed to be  $X_{01} = (2,5)$ . As a result of optimization  $X_{09} = (6,2)$  is obtained. The BESM-3m computer required 40 min machine time for founding  $Q_{\min}(X)$ .

In conclusion, the author thanks V.M. Faivyshevski for valuable advice.

#### LITERATURE CITED

1. B.A. Rosenfeld, Multidimensional Spaces [in Russia], izd Nauka, Moscow, 1966.
2. N.P. Buslenko, Mathematical Modeling of Industrial Processes by Digital Computers [in Russian], izd Nauka, Moscow, 1964.
3. Recommendations on Mathematical Modeling of Machine Stock Utilization Processes [in Russia], izd. NIISP Gosstroya UkrSSR, Kiev, 1966.