

A FINITE-BUFFER QUEUE WITH A SINGLE VACATION POLICY: AN ANALYTICAL STUDY WITH EVOLUTIONARY POSITIONING

MARCIN WOŹNIAK *, WOJCIECH M. KEMPA *, MARCIN GABRYEL **, ROBERT K. NOWICKI **

* Institute of Mathematics
Silesian University of Technology, ul. Kaszubska 23, 44-100 Gliwice, Poland
e-mail: {marcin.wozniak, wojciech.kempa}@polsl.pl

**Institute of Computational Intelligence
Częstochowa University of Technology, Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
e-mail: {marcin.gabryel, robert.nowicki}@iisi.pcz.pl

In this paper, application of an evolutionary strategy to positioning a $GI/M/1/N$ -type finite-buffer queueing system with exhaustive service and a single vacation policy is presented. The examined object is modeled by a conditional joint transform of the first busy period, the first idle time and the number of packets completely served during the first busy period. A mathematical model is defined recursively by means of input distributions. In the paper, an analytical study and numerical experiments are presented. A cost optimization problem is solved using an evolutionary strategy for a class of queueing systems described by exponential and Erlang distributions.

Keywords: finite-buffer queue, evolutionary strategy, object positioning, queueing system, busy period, idle time, single vacation.

1. Introduction

Applications of finite-buffer queueing systems (QSs) in various areas of technical and economical sciences are evident. In telecommunications, they are widely used in modeling input/output traffic of data packets in Internet routers, LAN switches and DNS or DHCP servers. QSs help to manage data traffic and connections routing that are passed between workstations, clients and servers. In management, they can be applied in investigation of particular stages of manufacturing processed like accumulation in a magazine or transitions between successive stages of production processes. QSs help to simulate and optimize manufacturing from production to delivery. They help to model and optimize the process in examined real-life situations. One can use them to simulate and position production lines in factories or warehouse processes. In transport and logistics, they can be helpful in modeling transport stages. QSs make it easier to plan the route and necessary pit-stops to load a new portion or unload some parts of delivery. Queuing models are also important in security systems, fire alarms and many other detection systems that must detect, predict or prevent danger. In fire alarms they help to optimize the

process of data acquisition from heat or smoke sensors placed in factories, markets or offices. Therefore, for positioning we need best possible methods.

In this article we present an attempt to apply computational intelligence (CI), particularly the evolutionary strategy (ES), in the process of positioning QSs. The presented research results may help us to compare various methods of CI and indicate which of them are tentatively best for positioning QSs. As an object for modeling and positioning, a $GI/M/1/N$ finite buffer QS with an exhaustive service and a single vacation policy is chosen.

Apparently due to emerging practical applications, queueing models with different-type restrictions of service processes are particularly significant. In the article, a system with a vacation policy is considered. In particular, we consider a single vacation policy with exhaustive service. In this type of QS we have a randomly distributed vacation (during which the service process is blocked) only when the system empties at the end of a busy period. Analyzing QSs with vacations can be useful in modeling many real-life problems like server maintenance in computer networks or temporary

restrictions in LAN traffic or road traffic in transport and logistics.

Since the paper of Doshi (1986), where an overview of vacation models was presented, many articles devoted to different types of service restrictions have published. As one can notice, a majority of results concern the system's characteristics in a stationary state. However, analysis of stochastic characteristics of QS in a transient (non-stationary) state is often necessary. For example, due to permanent changing parameters of Internet traffic, the stationary state in practice does not occur. In a LAN, clients are connecting and disconnecting simultaneously. Data packets are of different sizes and types. Moreover, clients may send many various requests at each moment. Therefore, this situation makes it important to investigate QSs of different types in transient mode.

In the paper, we present a $GI/M/1$ -type QS with finite buffer, a single vacation policy and exhaustive service. Some interesting examples of QSs with exhaustive service and a single vacation are presented by Takagi (1993; 1994). Vacation cycles in a QS under limited discipline are discussed in detail by Gupta *et al.* (2005) as well as Gupta and Sikdar (2006), while a QS with batch Markovian arrival processes is discussed by Niu and Takahashi (1999) (for finite capacity QSs with exhaustive vacation) and Niu *et al.* (2003) (for QSs with additional setup and close-down times).

An obvious use of queue models and Markov chains is modelling, analyzing and improving communication and network systems. A Markov chain is also applied to modeling biological systems such as phytoplankton in oceans (Wieczorek, 2010). Application of a Markov dynamic system in island modeling is discussed by Schaefer *et al.* (2012).

In this paper, using the embedded Markov chain approach and the formula of total probability, we find an explicit formula for transforms of first vacation cycle characteristics: durations of a busy period, idle time and the number of packets completely served during the first cycle. All formulas are written using a sequence defined recursively by means of the system's input parameters. Which representations can be efficiently used in numerical computations, as it was done in presented evolutionary positioning, can be included in the cost of work optimization. For a class of systems described by exponential and Erlang distributions, we find an optimal set of system parameters using an approach based on ES.

Infinite-buffer QSs with a vacation policy were considered by Choudhury (2002) as well as Kim and Yang (2011). In the work of Choudhury (2002), a formula for steady-state busy period duration was found. Kim and Yang (2011) derive a transform of a joint distribution function of busy period duration, number of packets completely served during a busy period and residual interarrival time at the end of the busy period

for a $GI/M/1$ system with phase-type vacations. The $M/G/1/b$ queuing system for analyzing the dropping function in an Internet router was used by Chydzinski and Chróst (2011).

In the work of Lee (1984), a steady-state busy period distribution was found for a finite-buffer $M/G/1$ -type queue with vacations and exhaustive service. Transient results for QSs with server vacations are rather rare. In the works of Kempa (2004; 2010c; 2010a; 2010b; 2010d; 2011b), explicit representations for the 2-fold transform of a departure counting process in an infinite-buffer $M^X/G/1$ -type system with different kinds of server vacations were derived. The transient queue-size distribution in a system with an N -policy, multiple vacations and setup times was investigated by Kempa (2010b). A formula for the transform of a joint distribution function of first vacation cycle components in the system with an infinite buffer were obtained by Kempa (2009; 2010d) using two different approaches. Transient results for distributions of the busy period and the idle time in a system without limitations in service can be found, e.g., in the works of Bratiichuk (2000) (finite-buffer case) as well as Bratiichuk and Kempa (2003) (infinite-buffer case). A non-stationary analysis of the virtual waiting time distribution in an $M/G/1/N$ -type system with an finite buffer and single vacations is given by Kempa (2011b).

The idea of cost control in QSs was considered by Teghem (1986), who introduced a classical linear cost structure. The optimization cost problem was analyzed by Kella (1990), Lillo (2000) and Teghem (1987), but for the Poisson arrivals model only. In particular, in the work of Lillo (2000) the existence of an optimal policy was proved for a system with vacations and exhaustive service. Piórkowski and Werewka (2010) try to optimize the process of packet completion in asynchronous communication. Woźniak *et al.* (2013) discuss efficient NoSQL system positioning. Representation for a total expected cost per time unit in the stationary state of a $M/G/1$ -type QS with a removable server and a finite buffer was obtained by Teghem (1987). One can find new results for the transient departure process in the $M^X/G/1$ infinite-buffer QS with different-type server vacations e.g., in the works of Kempa (2010c; 2011a; 2013), who also gives explicit representations for Laplace transforms of queue-size distribution in models with some mixed vacation policies (Kempa, 2012a). Characteristics of a vacation cycle were investigated also by Kempa (2009; 2010a), who additionally analyzed the queuing delay in a finite-buffer queue with single server vacations (Kempa, 2012b).

In this paper we present a possible way to position QSs using the evolutionary strategy (ES). Evolutionary algorithms are a simple and effective way to simulate and position examined objects. Computational intelligence, in particular, evolutionary or genetic algorithms, can

easily imitate the behavior of real life organisms applied in processes of optimization or data acquisition. As described by Nowak and Woźniak (2008a), Cpałka (2009), Garyel *et al.* (2012; 2013), Woźniak (2013) or Woźniak *et al.* (2014), CI can efficiently help in the analysis and positioning of various types of objects. In the work of Gabryel *et al.* (2012), application of an evolutionary algorithm to modeling and data acquisition in decision support systems was described. Nowak and Woźniak (2008a) or Gabryel *et al.* (2012) describe application of the genetic method to create learning sets for control systems. Nowak and Woźniak (2008b; 2009) discuss application of EC methods to positioning dynamic mechanical systems.

Using these methods one can not only examine the positioned object but also optimize values of input parameters, which is very important in optimization. Genetic or evolutionary algorithms enable testing objects in conditions similar to normal or routine work. They also help in analyzing extreme situations. We can easily adopt CI to imitate a situation of the highest traffic or the exhaustive service and therefore without problems or constructing sophisticated mechanisms analyze the examined object (here the queueing system) for the best possible positioning. Even if calculation describes the situation of destroying an object, we do not destroy a real one. We simply apply corrections that come from an analysis of numerical experiments into the model and test it again. Evolutionary computation methods are easy to implement and efficient in computations. The results of these operations are of good accuracy. Therefore, this is not only an easy, efficient but also less complicated way of positioning modeled object. That is why we would like to present application of the ES to the analyzing and positioning of modeled $G/M/1/N$ -type QSs.

2. Queueing model

In the paper we consider a finite-buffer $GI/M/1/N$ -type queueing system in which interarrival times are generally distributed random variables with a distribution function $F(\cdot)$ and service times have exponential distributions with mean μ^{-1} . The maximal system capacity equals N : we have $(N - 1)$ places in the buffer and one place in service. We assume that the system starts working at $t = 0$ with at least one packet present. After each busy period the server takes compulsory single vacation, with general-type distribution function $V(\cdot)$, during which the service process is blocked. If at the end of vacation the system is still empty, then the server is activated (is in standby mode) and “waits” for the first arrival. If vacation ends with some packets present in the system, then the service process is initialized immediately and a new busy period begins. We assume that sequences of successive interarrival times, service times and single vacation

duration are totally independent random variables.

Let us introduce the following notation:

- τ_1 : the first busy period of the system (starting at $t = 0$);
- δ_1 : the first idle time of the system (consisting of the first vacation time v_1 and the first server standby time q_1);
- $h(\tau_1)$: the number of packets completely served during τ_1 ;
- $X(t)$: the number of packets present in the system at time t .

In Fig. 1 we can see an example of time-line operation of the system. The modeled QS starts working with some packets present in the buffer. QS responds to given requests and the number of packets is changing in time. This period, when the server handles packets continuously, is called a busy period and is marked with τ_1 . Then we have a vacation time, marked with v_1 , when service in the system is stopped. Next is q_1 , when the system is gathering requests to serve. These two periods, v_1 and q_1 , build the idle time. Then a new busy period starts. Each busy period τ_1 and idle time δ_1 compose the operation period in the modeled QS, whose cost will be modeled and optimized.

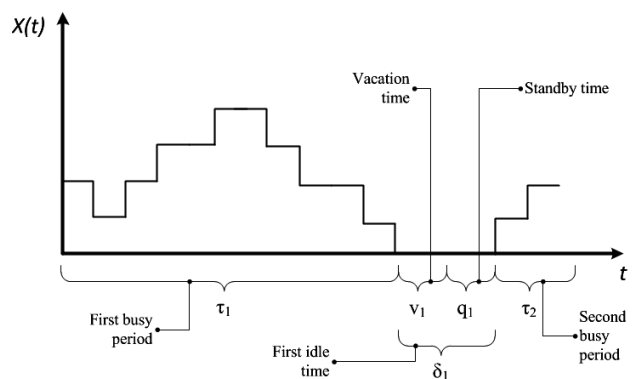


Fig. 1. Example of time-line operation of the modeled QS.

Let us now discuss an analytical model of these operations. In the works of Korolyuk (1975) and Korolyuk *et al.* (1987), the following system of equations is considered:

$$\sum_{k=-1}^n a_{k+1} x_{n-k} - x_n = \psi_n, \quad n \geq 0, \quad (1)$$

where (a_n) , $n \geq 0$, ($a_0 \neq 0$) and (ψ_n) , $n \geq 0$ are known sequences and the sequence (x_n) , $n \geq 0$, is unknown. As it turns out, an explicit representation for x_n can be found

and written in terms of a sequence defined recursively using the sequence (a_n) in the following way:

$$R_0 = 0, \quad R_1 = a_0^{-1},$$

$$R_{n+1} = R_1(R_n - \sum_{k=0}^n a_{k+1}R_{n-k}), \quad n \geq 1. \quad (2)$$

According to Korolyuk (1975) and Korolyuk *et al.* (1987), (R_n) is called the potential of (a_n) . In this article, we are interested in an explicit formula for a conditional joint characteristic function of the random variables τ_1, δ_1 and $h(\tau_1)$ defined as follows:

$$B_n(s, \varrho, z) = \mathbf{E}\{e^{-s\tau_1 - \varrho\delta_1} z^{h(\tau_1)} \mid X(0) = n\}, \quad (3)$$

where $1 \leq n \leq N, s \geq 0, \varrho \geq 0$ and $|z| \leq 1$. Using the formula of total probability written with respect to the first arrival epoch after the opening of the system, we will build a system of integral equations for $B_1(s, \varrho, z), \dots, B_N(s, \varrho, z)$. To find a solution, we use results from Theorem 1.

Theorem 1. *The solution of Eqn. (1) can be written explicitly in the following form:*

$$x_n = CR_{n+1} + \sum_{k=0}^n R_{n-k}\psi_k, \quad n \geq 0, \quad (4)$$

where C is a constant independent of n and (R_n) is defined in Eqn. (2).

Proof. Below we give a sketch of the proof only. For more details, please refer to Korolyuk (1975). Firstly, the operator K is introduced, which is defined on an arbitrary sequence (θ_n) as follows:

$$K\{\theta_n\} = \sum_{k=-1}^n a_{n-k}\theta_{n-k} - \theta_n.$$

It is easy to note that now the system (1) can be rewritten in the form

$$K\{x_n\} = \psi_n, \quad n \geq 0.$$

Next it is shown that, by definition, $K\{R_k\} = 0$, and moreover,

$$K\left\{\sum_{k=0}^n R_{n-k}\psi_k\right\} = \psi_n$$

which proves the formula (4). Finally, it is proved that any solution of (1) has the form (4). Indeed, writing (1) in another form, we get

$$x_{n+1} = a_0^{-1}\left(\psi_n + x_n - \sum_{k=0}^n a_{k+1}x_{n-k}\right), \quad n \geq 0,$$

so any sequence (x_n) satisfying (1) is uniquely determined by its first “zero” term, namely, x_0 . From (4) it follows that $x_0 = CR_1$, so, in consequence, it is possible to obtain any $x_0 \in \mathbb{R}$ by taking a proper C , i.e., $C = R_1^{-1}x_0$. ■

After operations on subscripts, the corollary below follows from Theorem 1.

Corollary 1. *The general solution of the following system of equations:*

$$\sum_{k=-1}^{n-2} a_{k+1}x_{n-k} - x_n = \psi_n, \quad n \geq 2, \quad (5)$$

can be written as

$$x_n = CR_{n-1} + \sum_{k=2}^n R_{n-k}\psi_k, \quad n \geq 2, \quad (6)$$

where we use the notation introduced above, C does not depend on n and (R_n) is defined in Eqn. (2).

2.1. Main analytical result. The main goal of this section is a theorem below, which gives the explicit representation for the joint transform $B_n(s, \varrho, z), n = 1, \dots, N$, of the distribution functions of the first busy period τ_1 , the first idle time δ_1 and the number $h(\tau_1)$ of the packets successfully processed during the first busy period τ_1 in the queueing system considered, conditioned by the initial “state” of the buffer. In the proof, the embedded Markov chain paradigm, the total probability law and the results from Corollary 1 will be applied.

Theorem 2. *For $B_n(s, \varrho, z), n = 1, \dots, N$, the following formulas are true:*

$$B_1(s, \varrho, z) = \mathbf{E}\{e^{-s\tau_1 - \varrho\delta_1} z^{h(\tau_1)} \mid X(0) = 1\}$$

$$= \frac{D(s, \varrho, z) - G(s, \varrho, z)}{H(s, z)} - \Psi_1(s, \varrho, z) \quad (7)$$

and

$$B_n(s, \varrho, z) = \mathbf{E}\{e^{-s\tau_1 - \varrho\delta_1} z^{h(\tau_1)} \mid X(0) = n\}$$

$$= \frac{D(s, \varrho, z) - G(s, \varrho, z)}{H(s, z)}R_{n-1}(s, z)$$

$$+ \sum_{k=2}^n R_{n-k}(s, z)\Psi_k(s, \varrho, z), \quad (8)$$

$2 \leq n \leq N$, where $s \geq 0, \varrho \geq 0, |z| \leq 1$, and

$$R_0(s, z) = 0,$$

$$R_1(s, z) = a_0^{-1}(s, z),$$

$$R_{n+1}(s, z) = R_1(s, z)(R_n(s, z) - \sum_{k=0}^n a_{k+1}(s, z)R_{n-k}(s, z)), \quad (9)$$

$$\begin{aligned} \Psi_n(s, \varrho, z) &= -\frac{(z\mu)^n}{(n-1)!} \left[\int_0^\infty dF(t) \int_0^t x^{n-1} e^{-(\mu+s)x} \right. \\ &\quad \left. \times \left(e^{-\varrho(t-x)} V(t-x) + \int_{t-x}^\infty e^{-\varrho y} dV(y) \right) dx \right], \end{aligned} \quad (10)$$

$$\begin{aligned} D(s, \varrho, z) &= \sum_{k=1}^{N-1} a_k(s, z) \sum_{i=2}^{N-k+1} R_{N-k+1-i}(s, z) \Psi_i(s, \varrho, z), \end{aligned} \quad (11)$$

$$\begin{aligned} G(s, \varrho, z) &= \Psi_N(s, \varrho, z) + (1 - f(\mu + s)) \\ &\quad \times \sum_{k=2}^N R_{N-k}(s, z) \Psi_k(s, \varrho, z), \end{aligned} \quad (12)$$

$$\begin{aligned} H(s, z) &= (1 - f(\mu + s)) R_{N-1}(s, z) \\ &\quad - \sum_{k=1}^{N-1} a_k(s, z) R_{N-k}(s, z), \end{aligned} \quad (13)$$

where the sequence $(a_n(s, z))$ is defined in the following way:

$$a_n(s, z) = \int_0^\infty \frac{(z\mu t)^n}{n!} e^{-(\mu+s)t} dF(t), \quad (14)$$

Proof. As is well known (see, e.g., Cohen, 1982), arrival epochs are Markov moments in the $GI/M/1$ -type queue. From the law of total probability applied with respect to the first arrival moment t after opening the system, we obtain the following formula:

$$\begin{aligned} B_n(s, \varrho, z) &= \sum_{k=0}^{n-1} B_{n-k+1}(s, \varrho, z) z^k \int_0^\infty \frac{(\mu t)^k}{k!} e^{-(\mu+s)t} dF(t) \\ &\quad + \frac{(z\mu)^n}{(n-1)!} \int_0^\infty dF(t) \int_0^t x^{n-1} e^{-(\mu+s)x} \\ &\quad \times \left[e^{-\varrho(t-x)} V(t-x) + \int_{t-x}^\infty e^{-\varrho y} dV(y) \right] dx, \end{aligned} \quad (15)$$

where $1 \leq n \leq N - 1$.

Let us comment briefly on Eqn. (15). The first summand on its right-hand side relates to a situation in which there are some completed services before the first arrival epoch t , but the buffer does not empty completely. If the number of served packets equals k , then at the Markov moment t the system evolution “renews” with $n - k + 1$ packets present. In the second summand on the right-hand side of Eqn. (15) the buffer empties at time

$x < t$. If vacation also ends before the first arrival epoch t , then the first cycle ends precisely at t . If vacation duration y exceeds $t - x$, then the busy period duration equals x and the idle time equals y . If the buffer is saturated at the opening ($X(0) = N$), we obtain

$$\begin{aligned} B_N(s, \varrho, z) &= B_N(s, \varrho, z) \int_0^\infty e^{-(\mu+s)t} dF(t) \\ &\quad + \sum_{k=1}^{N-1} B_{N-k+1}(s, \varrho, z) z^k \int_0^\infty \frac{(\mu t)^k}{k!} e^{-(\mu+s)t} dF(t) \\ &\quad + \frac{(z\mu)^N}{(N-1)!} \int_0^\infty dF(t) \int_0^t x^{N-1} e^{-(\mu+s)x} \\ &\quad \times \left[e^{-\varrho(t-x)} V(t-x) + \int_{t-x}^\infty e^{-\varrho y} dV(y) \right] dx. \end{aligned} \quad (16)$$

Interpretation of the right-hand side of Eqn. (16) is similar to that of Eqn. (15). The only difference is in the situation in which there are no departures before the first arrival epoch t . Then the packet occurring at time t is lost due to buffer saturation and the number of packets present in the system at the Markov moment t equals N (cf. the first summand on the right-hand side of Eqn. (16)).

Introducing $a_n(s, z)$ as in Eqn. (14), we can transform Eqns. (15)–(16) into the following form:

$$\begin{aligned} \sum_{k=-1}^{n-2} a_{k+1}(s, z) B_{n-k}(s, \varrho, z) - B_n(s, \varrho, z) &= \Psi_n(s, \varrho, z), \end{aligned} \quad (17)$$

for $1 \leq n \leq N - 1$ and

$$\begin{aligned} B_N(s, \varrho, z) (1 - f(\mu + s)) &= \sum_{k=1}^{N-1} a_k(s, z) B_{N-k+1}(s, \varrho, z) - \Psi_N(s, \varrho, z), \end{aligned} \quad (18)$$

where

$$f(s) = \int_0^\infty e^{-st} dF(t), \quad s > 0, \quad (19)$$

and $\Psi_n(s, \varrho, z)$ was defined in Eqn. (10).

Let us note that Eqn. (17) has the same form as Eqn. (5). Thus, a solution of Eqn. (17) can be found using the result from Eqn. (6). Of course, now C is, in general, a function of s, ϱ and z , where the potential $(R_n(s, z))$ can be found as in Eqn. (9), using the recurrent formula in Eqn.(2) with $a_n(s, z)$ defined in Eqn. (14) instead of (a_n) . From Eqn. (6) we obtain

$$\begin{aligned} B_n(s, \varrho, z) &= C(s, \varrho, z) R_{n-1}(s, z) \\ &\quad + \sum_{k=2}^n R_{n-k}(s, z) \Psi_k(s, \varrho, z), \end{aligned} \quad (20)$$

where $k = 2, \dots, n$.

Now we should find representations for $C(s, \rho, z)$ and $B_1(s, \rho, z)$. Substituting $n = 1$ into Eqn. (17) we have

$$B_1(s, \rho, z) = a_0(s, z)B_2(s, \rho, z) - \Psi_1(s, \rho, z). \quad (21)$$

Similarly, substituting $n = 2$ into Eqn. (20), we obtain

$$B_2(s, \rho, z) = C(s, \rho, z)R_1(s, z) = \frac{C(s, \rho, z)}{a_0(s, z)}. \quad (22)$$

Substituting now Eqn. (22) into Eqn. (21), we get

$$B_1(s, \rho, z) = C(s, \rho, z) - \Psi_1(s, \rho, z). \quad (23)$$

To find an explicit formula for $C(s, \rho, z)$, let us implement Eqn. (20) into Eqn. (18). We have

$$\begin{aligned} & \left[C(s, \rho, z)R_{N-1}(s, z) + \sum_{k=2}^N R_{N-k}(s, z)\Psi_k(s, \rho, z) \right] \\ & \times (1 - f(\mu + s)) \\ & = \sum_{k=1}^{N-1} a_k(s, z) \times \left[C(s, \rho, z)R_{N-k}(s, z) \right. \\ & \left. + \sum_{i=2}^{N-k+1} R_{N-k+1-i}(s, z)\Psi_i(s, \rho, z) \right] - \Psi_N(s, \rho, z), \end{aligned} \quad (24)$$

and hence we eliminate $C(s, \rho, z)$ as follows:

$$C(s, \rho, z) = \frac{D(s, \rho, z) - G(s, \rho, z)}{H(s, z)}, \quad (25)$$

where $D(s, \rho, z)$, $G(s, \rho, z)$ and $H(s, z)$ were defined in Eqns. (11), (12) and (13), respectively. Now we complete the proof by collecting Eqns. (20), (23) and (25). ■

Let us note that, from the last theorem, a formula for conditional means of the first busy period τ_1 , the first idle time δ_1 and the number of departures occurring during τ_1 separately can be found.

Indeed, since

$$\mathbf{E}_n e^{-s\tau_1} = \mathbf{E}\{e^{-s\tau_1} | X(0) = n\} = B_n(s, 0, 1), \quad (26)$$

we get

$$\mathbf{E}_n \tau_1 = -\frac{\partial}{\partial s} B_n(s, 0, 1) \Big|_{s=0}. \quad (27)$$

Similarly, we have

$$\mathbf{E}_n \delta_1 = -\frac{\partial}{\partial \rho} B_n(0, \rho, 1) \Big|_{\rho=0} \quad (28)$$

and

$$\mathbf{E}_n h(\tau_1) = \frac{\partial}{\partial z} B_n(0, 0, z) \Big|_{z=1}. \quad (29)$$

The QS model presented in this section was solved using Wolfram Mathematica 9.0. It helped us to prepare appropriate analytical forms of Eqns. (27)–(29) describing service, vacation and income.

3. Research results

In our investigatious and QS modeling for ES positioning and analysis, we considered the function

$$F(t) = 1 - (1 + \lambda \cdot t) \cdot e^{-\lambda \cdot t} \quad (30)$$

to describe packet interarrival times. We assumed the probability distribution of interarrival times in the system (the distance in time between successive packets coming into the node of a sensor network) to be a 2-Erlang distribution with parameter λ , which means that interarrival times are indeed random and the average interarival time is $2/\lambda$. We also considered the function

$$V(t) = 1 - e^{-\alpha \cdot t} \quad (31)$$

to describe server compulsory vacation time when service process is stopped. $V(\cdot)$ is a distribution function of vacation time (no transmission time) in the system. We assumed that the silent time (single vacation) has an exponential distribution with parameter α , which is random and on the average amounts to $1/\alpha$. Therefore, $V(\cdot)$ is distribution duration of the suspension manual (transmission) by the node. In the works of Hongwei et al. (2010) as well as Mancuso and Alouf (2012) similar systems are considered, in which periods of energy saving of the node in a wireless network (they are called “sleep modes”) are modeled using such a repetitive (renewable) period of suspension (the period is renewed when there are still no packets). For example, in the work of Hongwei et al. (2010) a model of a sensor network with five nodes and one node container (sink node) is considered, while Mancuso and Alouf (2012) discuss a model of vacations for a node in power saving mode defined by 3GPP.

Using functions from Eqns. (30) and (31), one can estimate the potential cost of system work. Let us now analyze Eqns. (27)–(29). In Figs. 2–5 charts of the examined values are presented. In the research and analysis, n was taken as unity. The first busy period of the system can be analyzed using $\mathbf{E}_1 \tau_1$ in the function of μ based on Eqn. (27) and results are plotted in Fig. 2. These values describe the expected work of the system with fixed times of interarrivals and breaks. One can estimate that with an increasing parameter μ , corresponding to service time, the analyzed QS may earlier start a longer busy period.

The first idle time can be analyzed using $\mathbf{E}_1 \delta_1$ as a function of α based on Eqn. (28), and the results are plotted in Fig. 3. These values describe the expected work of the system with fixed times of income and service. One can estimate that with increasing parameter α , corresponding to service time, the analyzed QS may earlier start a longer vacation period.

The first idle time can be analyzed using $\mathbf{E}_1 \delta_1$ in the function of λ based on Eqn.(28), and our results

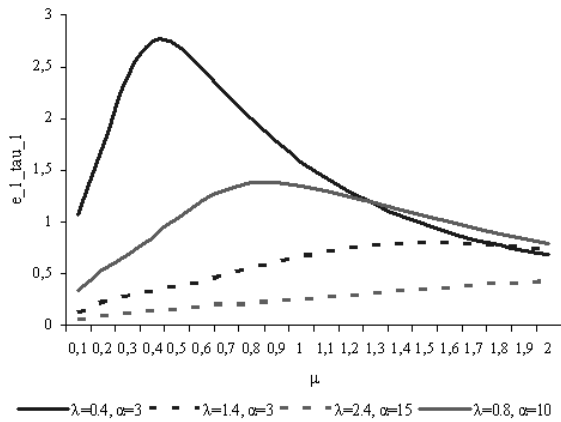


Fig. 2. $E_1 \tau_1$ as a function of μ .

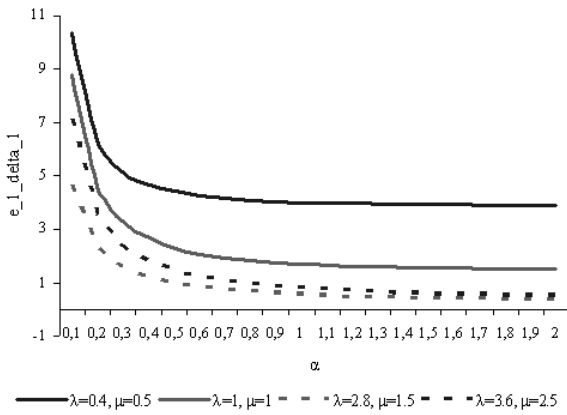


Fig. 3. $E_1 \delta_1$ as a function of α .

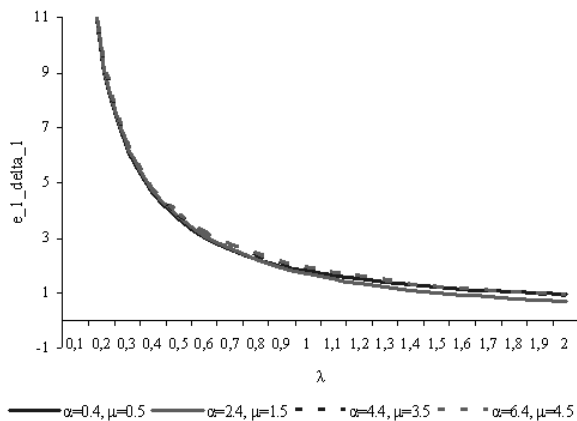


Fig. 4. $E_1 \delta_1$ as a function of λ .

are plotted in Fig. 4. These values describe the expected income into the system with fixed times of vacation and service. Comparing the results, we can estimate that for smaller parameters α , corresponding to the vacation time, or λ , corresponding to the interarrival times, the QS should serve more incoming packets in busy periods.

The number of packets completely served can be analyzed using $E_1 h(\tau_1)$ in the function of λ based on Eqn. (28), and the results are plotted in Fig. 5. These

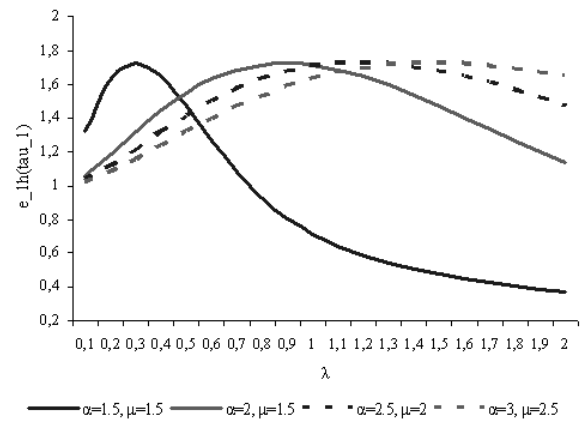


Fig. 5. $E_1 h(\tau_1)$ from Eqn. (29) as a function of λ .

values describe the expected income into the system with fixed times of vacation and service times. One can estimate that the parameter λ , corresponding to the income time, is very important for positioning service and vacation times in the modeled QS.

Let us also analyze $E_1 h(\tau_1)$ from Eqn. (28) as a function of μ . Results are plotted in Fig. 6. One

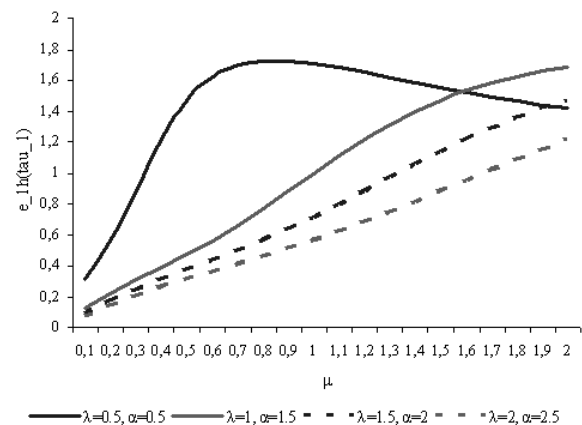


Fig. 6. $E_1 h(\tau_1)$ from Eqn. (29) as a function of μ .

can estimate that with increasing the parameter μ ,

corresponding to the service time, more requests can be completely served. The parameter λ , corresponding to interarrival times, may also have some influence on the system. For lower values of λ and medium interarrival or vacation times, the system may serve more requests. With higher values of λ and lower interarrival or vacation times, there is a possibility to serve fewer requests. However, positioning for the lowest total cost of service is the main problem for every QS. Therefore, having analyzed and modeled an $E_2/M/1/N$ -type system, we will try to position it for the best possible service at the lowest cost.

3.1. QS cost of service: The optimization problem. The optimal cost of work is very important in any system. In our research we define it to describe the minimal amount of resources to perform all operations we need.

Definition 1. The *optimal cost of work* in a modeled QS is the minimal amount of money, energy or any other resources that the system may need to perform all operations.

The optimal cost of work of Definition 1 will have the form of an equation to be used in the optimization process. The classic, linear cost structure was introduced by Teghem (1986). The optimization cost problem was also analyzed by Kella (1990), Lillo (2000) and Teghem (1987), but for the Poisson arrival model only. Here we present formulas for cost optimization for a class of QSs described by exponential and Erlang distributions (Eqns. (30) and (31), respectively).

The equation representing the total cost of work for the system during the first vacation cycle c_1 , the first busy period τ_1 followed by the first idle time δ_1 is

$$Q_n(c_1) = r(\tau_1)\mathbf{E}_n\tau_1 + r(\delta_1)\mathbf{E}_n\delta_1, \quad (32)$$

where $Q_n(c_1)$ is the cost of operation during the first cycle c_1 on the condition that $X(0) = n$, $r(\tau_1)$ are fixed unit costs of operation during the first busy period τ_1 , $r(\delta_1)$ are fixed unit costs of operation during the first idle time δ_1 , $\mathbf{E}_n\tau_1$ is the mean duration of the first busy period conditioned by $X(0)$, $\mathbf{E}_n\delta_1$ the mean duration of the first idle period conditioned by $X(0)$. The symbols $r(\tau_1) = r_{\tau_1}$ and $r(\delta_1) = r_{\delta_1}$ are fixed costs of operation for modeled QS.

The notation in Eqn. (32) is similar to describing cost functions of time. However, we write in this way to make the article clearer and not to complicate symbols. The unit cost of operation during the first cycle c_1 , on condition that primarily the system contains exactly n packets, can be expressed as

$$r_n(c_1) = \frac{Q_n(c_1)}{\mathbf{E}_n(c_1)} = \frac{r(\tau_1)\mathbf{E}_n\tau_1 + r(\delta_1)\mathbf{E}_n\delta_1}{\mathbf{E}_n\tau_1 + \mathbf{E}_n\delta_1}, \quad (33)$$

where $r_n(c_1)$ is the total cost of service, $Q_n(c_1)$ is the cost of service, $\mathbf{E}_n(c_1)$ is the mean duration of the first cycle, and the other symbols are the same as in Eqn. (32). It denotes modeling a total service cost of the system.

In our research, for ES simulation and positioning we assumed $r(\tau_1) = 0.1$ and $r(\delta_1) = 0.1$. However, these values depend on the examined system and since the model we present in Section 2 is general, for a $GI/M/1/N$ -type finite-buffer QS one can have other values to describe the system in examination and design.

Positioning is a very important phase in the process of designing a QS. To perform this operation we must know possible malfunctions and optimal work conditions. We often apply knowledge that comes from human experts or our previous experience. However, the best way is to perform computer simulations using an efficient method. We will solve optimization problem using the ES described in detail by Eiben and Smith (2003) or Gabryel and Rutkowski (2010).

3.2. ES for QS cost optimization. ES belongs to a general class EC methods. Apart from the ES, EC comprises genetic algorithms, evolutionary programming and heuristic methods, too. ES applied in the presented QS positioning is an optimization technique based on adaptation and evolution of input data to find optimum modeled objects in the search space (see Eiben and Smith, 2003).

There exist two main forms of the evolutionary strategy, namely, $(\mu' + \lambda')$ ES and (μ', λ') ES, where μ' denotes the number of parents and λ' stands for the number of all offspring created by parents within one generation. Parents are deterministically selected out of either offspring, referred to as comma-selection ($\mu' < \lambda'$ must be hold), or both parents and offspring, referred to as plus-selection. The symbols μ and λ , although similar to those in Section 2, have a different meaning here. Similar notation is just coincidence, because of their present implementations in the theory of QS and EC.

Individuals are “reproduced” in the next generation by random selection of λ' individuals. Selection is based on the ranking of individuals’ fitness (adaptation to given conditions) taking μ' best individuals. We draw and put selected ones in a temporary population. These individuals undergo crossover and mutation, resulting in a group of ancestors (containing also λ' individuals). Finally, we select the best μ' descendants, which will form a new parental population and the rest is taken at random in the search space.

The mutation process is based on the added chromosome σ , which contains the values of the standard deviations of each chromosome gene σ_i for $i = 1, \dots, L$, where L is the length of chromosome. Mutation is performed on a single individual. First we mutate the

chromosome σ according to the formula

$$\sigma'_i = \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)}, \quad (34)$$

where σ_i is the local mutation step size in the generation, σ'_i is the local mutation step size in the next generation, $N(0, 1)$ and $N_i(0, 1)$ are normally distributed random values, generated for chromosomes and genes, respectively, $\tau' = 1/\sqrt{2L}$, $\tau = 1/\sqrt{2\sqrt{L}}$, $i = 1, \dots, L$, where L is the chromosome length.

The crossing operator proceeds on both the chromosomes: the vector x of independent variables and the standard deviation σ . The crossover operation is drawing two individuals and exchanging or averaging values of their genes. Two new individuals replace their parents.

Let us briefly describe crossing. We choose two individuals $(x^1, \sigma^1) = ([x_1^1, \dots, x_n^1]^T, [\sigma_1^1, \dots, \sigma_n^1]^T)$ and $(x^2, \sigma^2) = ([x_1^2, \dots, x_n^2]^T, [\sigma_1^2, \dots, \sigma_n^2]^T)$. Its new individual is formed according to

$$(x')_i^1 = a \cdot x_i^1 + (1 - a) \cdot x_i^2, \quad (35)$$

$$(x')_i^2 = a \cdot x_i^2 + (1 - a) \cdot x_i^1, \quad (36)$$

$$(\sigma')_i^1 = a \cdot \sigma_i^1 + (1 - a) \cdot \sigma_i^2, \quad (37)$$

$$(\sigma')_i^2 = a \cdot \sigma_i^2 + (1 - a) \cdot \sigma_i^1, \quad (38)$$

where a is a number randomly sampled from the uniform distribution $U(0, 1)$.

Next, the chromosome x is changed according to

$$x'_i = x_i + \sigma'_i \cdot N_i(0, 1). \quad (39)$$

Finally, the standard evolution strategy is extended by making use of a uniform recombination operator (see Eiben and Smith, 2003; Michalewicz, 1996).

The simulation system is based on the mathematical model described in Section 2.1. Equation (33) was used to optimize the total cost of work (as the fitness function in ES). The ES simulation system was searching for the best values of the examined object that make it work with the lowest costs in specified conditions, as presented in Fig. 7. Our research provides a type of knowledge that describes an example system operation in some possible scenarios. This type of knowledge is necessary for tuning and evaluating the examined object.

3.3. ES simulation and positioning results. The mathematical model presented in Section 2 helps us to give analytical form of QS estimators representing the service time (Eqn. (27)) and the vacation time (Eqn. (28)) in the system. These equations are used to describe the total cost of service, defined in Eqn. (33), which is applied as the fitness function presented in Section 3.2 to position the examined QS.

Queueing system modeling: exponential and Erlang distributions to describe busy period, idle time, vacation

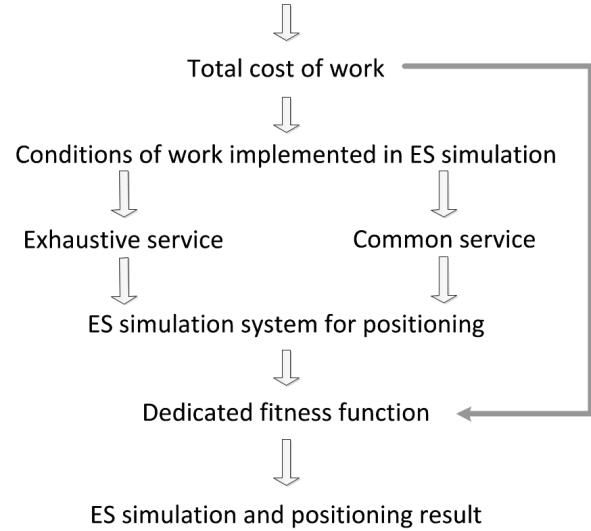


Fig. 7. Modeling for the ES simulation process.

Let us now present experimental results for the optimal cost of work of the examined $GI/M/1/N$ -type QS. The presented results are an average of 100 samplings in each operation of positioning. Optimal positioning and all presented scenarios were determined using ES simulation. In our research we assumed the population size $\mu' = 20$ and $\lambda' = 100$, with 100 generations (according to the notation in Section 3.2). We defined the following assumptions for parameters μ , λ and α (according to the notation in Section 2):

- average service time: $T_{\text{service}} = 1/\mu$,
- average time between packets' incomes into the system: $T_{\text{income}} = 2/\lambda$,
- average vacation time: $T_{\text{vacation}} = 1/\alpha$,
- examined system size: $N = \text{buffer size} + 1$, where buffer size = 5.

In Table 1 we present optimal values for all QS parameters μ , λ and α . However, we are able to set some values of the system we use. Therefore, we have also tried to optimize the values of parameters μ , λ and α in several possible scenarios. Each scenario was defined and then we optimized the values of the system parameters and the cost of work according to given assumptions. In each scenario, there were 100 ES experiments and results are given as the average of all of them. Let us now discuss the examined scenarios.

Scenario 1. In this scenario, to optimize the cost, we assumed that the system handles incoming packets in

Table 1. Optimal μ , λ and α for the lowest cost of work, cf. Eqn. (33).

Optimal QS parameter values	
μ	2.1
λ	0.05
α	13092.22
$r_n(c_1)$	0.01
Optimal time [sec]	
$T_{service}$	0.47
T_{income}	40.08
$T_{vacation}$	$7.63E^{-05}$

constant time. Thus the average service time $T_{service} = 1/\mu$ is constant. Therefore, we set the parameter μ and optimized other system parameters. In this scenario, ES simulation was done for the set values $\mu = 100$, $\mu = 1$ and $\mu = 0.01$. Experimental results are shown in Table 2.

Table 2. Optimal parameters λ and α for set $\mu = 100$, $\mu = 1$, $\mu = 0.01$ and the lowest cost value of Eqn. (33).

Optimal QS parameter values			
μ	100	1	0.01
λ	0.44	0.18	0.25
α	1.09	1.46	1.59
$r_n(c_1)$	0.01	0.09	12.91
Optimal time [sec]			
$T_{service}$	0.01	1	100
T_{income}	4.53	10.78	7.75
$T_{vacation}$	0.92	0.69	0.63

Scenario 2. In this scenario, to optimize the cost we assumed that packets come into the system with some regularity, and the time between packages income into the system, $T_{income} = 2/\lambda$, is constant. The parameter λ is set and the other system parameters are optimized. In this scenario, ES simulation was done for the set values $\lambda = 100$, $\lambda = 1$ and $\lambda = 0.01$. Experimental results are shown in Table 3.

Table 3. Optimal parameters μ and α for set $\lambda = 100$, $\lambda = 1$, $\lambda = 0.01$ and the lowest cost value of Eqn. (33).

Optimal QS parameter values			
μ	4.06	27.30	1.36
λ	100	1	0.01
α	$2.318E^{-06}$	0.001	1.5
$r_n(c_1)$	12.32	0.02	0.004
Optimal time [sec]			
$T_{service}$	0.27	0.04	0.74
T_{income}	0.02	2	200
$T_{vacation}$	431462.22	341997.27	0.67

Scenario 3. In this scenario, to optimize the cost we assumed that the system needs to stop serving requests

with some regularity; the vacation time, $T_{vacation} = 1/\alpha$, is constant and the parameter α is set. ES simulation was done for the set values $\alpha = 100$, $\alpha = 1$ and $\alpha = 0.01$. Experimental results are shown in Table 4.

Table 4. Optimal parameters μ and λ for set $\alpha = 100$, $\alpha = 1$, $\alpha = 0.01$ and the lowest cost value of Eqn. (33).

Optimal QS parameter values			
μ	1.17	1.25	1.23
λ	0.18	0.17	0.21
α	100	1	0.01
$r_n(c_1)$	0.08	0.07	0.09
Optimal time [sec]			
$T_{service}$	0.85	0.80	0.81
T_{income}	11.28	11.58	9.68
$T_{vacation}$	0.01	1	100

Moreover, we also analyzed some more complicated scenarios. We examined possible situations where the service time, the packets, income time or vacation time were set and the cost of work must be adequate.

Scenario 4. In this scenario, to optimize the cost, we assumed that service time $T_{service} = 1/\mu$, is constant. We set the parameter μ and optimized the other parameters. ES simulation was done for the set values $\mu = 100$, $\mu = 1$ or $\mu = 0.01$ and $r_n(c_1) < 1$ or $r_n(c_1) > 1$. Experimental results are shown in Tables 5 and 6.

Table 5. Optimal parameters α and λ for set μ and the lowest cost value of Eqn. (33) < 1 .

Optimal QS parameter values			
μ	100	1	0.01
λ	0.40	0.22	0.008
α	1.19	1.3	7.0
$r_n(c_1) < 1$	0.002	0.11	0.40
Optimal time [sec]			
$T_{service}$	0.01	1	100
T_{income}	5.0	8.94	249.58
$T_{vacation}$	0.84	0.77	0.14

Table 6. Optimal parameters α and λ for set μ and the lowest cost value of Eqn. (33) > 1 .

Optimal QS parameter values			
μ	100	1	0.01
λ	1310.42	16.56	0.23
α	0.001	0.001	1.48
$r_n(c_1) > 1$	6.55	8.28	11.63
Optimal time [sec]			
$T_{service}$	0.01	1	100
T_{income}	0.001	0.121	8.601
$T_{vacation}$	17682.2	2178649.2	0.68

Scenario 5. In this scenario, to optimize the cost, we assumed that the time between packages income into the system $T_{income} = 2/\lambda$, is constant. Therefore, we set λ and optimized the other parameters. In this scenario, ES simulation was done for the set values $\lambda = 100$, $\lambda = 1$ or $\lambda = 0.01$ and $r_n(c_1) < 1$ or $r_n(c_1) > 1$. Research results are shown in Tables 7 and 8.

Table 7. Optimal parameters α and μ for set λ and the lowest cost value of Eqn. (33) < 1 .

Optimal QS parameter values			
μ	1598.611	1.31	1.71
λ	100	1	0.01
α	0.003	0.18	1.36
$r_n(c_1) < 1$	0.03	0.07	0.03
Optimal time [sec]			
$T_{service}$	0.006	0.76	0.58
T_{income}	0.02	2	200
$T_{vacation}$	3593.54	10.84	0.74

Table 8. Optimal parameters α and μ for set λ and the lowest cost value of Eqn. (33) > 1 .

Optimal QS parameter values			
μ	4.78	0.29	0.002
λ	100	1	0.01
α	0.001	0.001	25.22
$r_n(c_1) > 1$	10.48	1.73	2.43
Optimal time [sec]			
$T_{service}$	0.21	3.47	485.4
T_{income}	0.02	2	200
$T_{vacation}$	310848.6	146370	0.04

Scenario 6. In this scenario, to optimize the cost we assumed that the vacation time, $T_{vacation} = 1/\alpha$, is constant. Therefore, we set α and optimized the other system parameters. For the cost of system work $r_n(c_1)$ defined in some way, all system parameters were optimized for $\alpha = 100$, $\alpha = 1$ or $\alpha = 0.01$ and $r_n(c_1) < 1$ or $r_n(c_1) > 1$. Experimental results are shown in Tables 9 and 10.

Table 9. Optimal parameters λ and μ for set α and the lowest cost value of Eqn. (33) < 1 .

Optimal QS parameter values			
μ	1.42	1.74	1.34
λ	0.21	0.11	0.18
α	100	1	0.01
$r_n(c_1) < 1$	0.08	0.03	0.07
Optimal time [sec]			
$T_{service}$	0.7	0.56	0.7
T_{income}	9.2	17.4	11.1
$T_{vacation}$	0.01	1	100

Table 10. Optimal parameters λ and μ for set α and the lowest cost value of Eqn.(33) > 1 .

Optimal QS parameter values			
μ	0.13	0.15	0.1
λ	0.5	0.48	0.46
α	100	1	0.01
$r_n(c_1) > 1$	1.73	1.6	2.23
Optimal time [sec]			
$T_{service}$	7.7	6.5	9.6
T_{income}	4.45	4.2	4.3
$T_{vacation}$	0.01	1	100

3.4. Conclusions. We applied dedicated ES to cost optimization and the positioning problem in a QS with a single vacation policy and exhaustive service. In the research we used EC methods, which are efficient in optimization (see Gabryel *et al.*, 2012; Woźniak, 2013).

Positioning gave results of optimal system parameters that can be applied to position a given QS for service at the lowest costs in different situations. We examined various situations that are common in reality, each defined by a special set of parameters. Numerical experiments gave proper results and confirmed that ES is easy to implement, precise and efficient.

The EC method presented in Section 2 is defined in general form, which makes it easily applicable to simulate and position other QSs of a similar type. To change the examined system, one may simply take other functions $F(\cdot)$ and $V(\cdot)$ defined in Eqns. (30) and (31), respectively, to model other similar type QSs.

The values of system parameters set in the scenarios defined in Section 3.3 presented various, miscellaneous but hypothetical cases of service. However, in real applications of the presented solution, system identification, i.e., determining values of its parameters and characteristics described in functions $F(\cdot)$ and $V(\cdot)$, becomes a nontrivial problem.

4. Final remarks

QSs with different type restrictions of service processes are particularly significant due to numerous practical applications. In the article, a QS with a single vacation policy and exhaustive service is considered. This type of QS is useful in modeling many real-life problems like server maintenance in computer networks, LAN and Wi-Fi traffic modeling, or road traffic simulation. It is also applicable in transport and logistics models.

In the article, we proposed an application of the ES for the modeled QS to simulation and positioning. However, other CI methods or cognitive algorithms may also be useful in simulation or positioning, or to generate collection of representative samples. Results can be used

by decision support systems or as knowledge for AI systems.

ES is very effective for positioning QSs, as shown in this paper. One may easily simulate and position examined object in different conditions. This method is useful when we have an analytical model of the object, which is very complex and mathematically complicated, where classic calculations are merely feasible. The conducted experiments show the efficiency of ES to simulate the system in many possible scenarios. An important restriction is to carry out a large number of experiments to determine the best possible description of the simulated object.

Further work should be carried out to reduce time consuming operations, tentatively by using some knowledge prior to generate an initial population in ES simulation and the positioning system. We will also try to implement other EC methods and examine their efficiency in simulation and positioning of modeled queueing systems.

Moreover, the modeled QS could be non-stationary and parameters could change during work due to, e.g., wear (expenditure) of elements and consumables (battery, oil, belts, etc.) or environment changes (temperature, air composition, etc.). So, in our future research we plan to take into account these aspects by, e.g., fuzzyfication of parameters in the modeled system identification (Rutkowski et al., 2012), or apply time-varying methods (Rutkowski, 2004) and their hybrid constructions (Jaworski et al., 2012).

References

- Bratiichuk, M.S. (2000). Exact formulas for $E^0/G/1/N$ -type queueing systems, *Ukrainian Mathematical Journal* **8**(52): 1034–1044.
- Bratiichuk, M.S. and Kempa, W.M. (2003). Application of the superposition of renewal processes to the study of batch arrival queues, *Queueing Systems* **1**(44): 51–67.
- Choudhury, G. (2002). A batch arrival queue with a vacation time under single vacation policy, *Computers and Operations Research* **13**(29): 1941–1955.
- Chydziański, A. and Chróst, Ł. (2011). Analysis of AQM queues with queue size based packet dropping, *International Journal of Applied Mathematics and Computer Science* **21**(3): 567–577, DOI: 10.2478/v10006-011-0045-7.
- Cohen, J. (1982). *The Single Server Queue*, North-Holland Publishing Company, Amsterdam.
- Cpałka, K. (2009). A new method for design and reduction of neuro-fuzzy classification systems, *IEEE Transactions on Neural Networks* **20**(4): 701–714.
- Doshi, B.T. (1986). Queueing systems with vacations—a survey, *Queueing Systems* **1**(1): 29–66.
- Eiben, A. and Smith, J. (2003). *Introduction to Evolutionary Computing*, Springer-Verlag, New York, NY.
- Gabryel, M., Nowicki, R.K., Woźniak, M. and Kempa, W. M. (2013). Genetic cost optimization of the $GI/M/1/N$ finite-buffer queue with a single vacation policy, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada (Eds.), *12th International Conference, ICAISC 2013, Zakopane, Poland, June 9–13, 2013, Proceedings, Part II*, Lecture Notes in Artificial Intelligence, Vol. 7895, Springer-Verlag, Berlin/Heidelberg, pp. 12–23.
- Gabryel, M. and Rutkowski, L. (2010). Evolutionary designing of logic-type fuzzy systems, in L. Rutkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada (Eds.), *10th International Conference, ICAISC 2010, Zakopane, Poland, June 13–17, 2010, Part II*, Lecture Notes in Artificial Intelligence, Vol. 6114, Springer-Verlag, Berlin/Heidelberg, pp. 143–148.
- Gabryel, M., Woźniak, M. and Nowicki, R.K. (2012). Creating learning sets for control systems using an evolutionary method, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada (Eds.), *International Symposia, SIDE 2012 and EC 2012, Held in Conjunction with ICAISC 2012, Zakopane, Poland, April 29–May 3, 2012, Proceedings*, Lecture Notes in Computer Science, Vol. 7269, Springer-Verlag, Berlin/Heidelberg, pp. 206–213.
- Gupta, U., Banik, A. and Pathak, S. (2005). Complete analysis of $MAP/G/1/N$ queue with single (multiple) vacation(s) under limited service discipline, *Journal of Applied Mathematics and Stochastic Analysis* **3**: 353–373.
- Gupta, U. and Sikdar, K. (2006). Computing queue length distributions in $MAP/G/1/N$ queue under single and multiple vacation, *Journal of Applied Mathematics and Computing* **2**(174): 1498–1525.
- Hongwei, D., Dongfeng, Z. and Yifan, Z. (2010). Performance analysis of wireless sensor networks of serial transmission mode with vacation on fire prevention, *2nd International Conference on Computer Engineering and Technology (IC-CET)*, Chengdu, China, Vol. 5, pp. 153–155.
- Jaworski, M., Duda, P. and Pietruczuk, L. (2012). On fuzzy clustering of data streams with concept DRIFT, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada (Eds.), *11th International Conference, ICAISC 2012, Zakopane, Poland, April 29–May 3, 2012, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 7268, Springer-Verlag, Berlin/Heidelberg, pp. 82–91.
- Kella, O. (1990). Optimal control of the vacation scheme in an $M/G/1$ queue, *Operations Research* **4**(38): 724–728.
- Kempa, W. (2004). The virtual waiting time for the batch arrival queueing systems, *Stochastic Analysis and Applications* **5**(22): 1235–1255.
- Kempa, W. (2009). $GI/G/1/\infty$ batch arrival queueing system with a single exponential vacation, *Mathematic Methods and Operations Research* **1**(69): 81–97.
- Kempa, W. (2010a). Characteristics of vacation cycle in the batch arrival queueing system with single vacations and

- exhaustive service, *International Journal of Applied Mathematics* **4**(23): 747–758.
- Kempa, W. (2010b). On departure process in the batch arrival queue with single vacation and setup time, *Annales UMCS: Informatica* **1**(10): 93–102.
- Kempa, W. (2010c). Some new results for departure process in the $M^X/G/1$ queueing system with a single vacation and exhaustive service, *Stochastic Analysis and Applications* **1**(28): 26–43.
- Kempa, W. (2010d). Some results for the actual waiting time in batch arrival queueing systems, *Stochastic Models* **3**(26): 335–356.
- Kempa, W. (2011a). Analysis of departure process in batch arrival queue with multiple vacations and exhaustive service, *Communications in Statistics—Theory and Methods* **40**(16): 2856–2865.
- Kempa, W. (2011b). Departure process in finite-buffer queue with batch arrivals, in K. Al-Begain, S. Balsamo, D. Fiems and A. Marin (Eds.), *18th International Conference, ASMTA 2011, Venice, Italy, June 20–22, 2011. Proceedings*, Lecture Notes in Computer Science, Vol. 6751, Springer-Verlag, Berlin/Heidelberg, pp. 1–13.
- Kempa, W. (2012a). On transient queue-size distribution in the batch arrival system with the n -policy and setup times, *Mathematical Communications* **17**(1): 285–302.
- Kempa, W. (2012b). The virtual waiting time in a finite-buffer queue with a single vacation policy, in K. Al-Begain, D. Fiems and J.-M. Vincent (Eds.), *Proceedings, 19th International Conference, ASMTA 2012, Grenoble, France, June 4–6, 2012*, Lecture Notes in Computer Science, Vol. 7314, Springer-Verlag, Berlin/Heidelberg, pp. 47–60.
- Kempa, W. (2013). Output process in batch-arrival queue with n -policy and multiple vacations, in A. Dudin and K. De Turck (Eds.), *20th International Conference, ASMTA 2013, Ghent, Belgium, July 8–10, 2013. Proceedings*, Springer-Verlag, Berlin/Heidelberg, Lecture Notes in Computer Science, Vol. 7314, pp. 247–261.
- Kim, K. and Yang, W.S. (2011). Busy period analysis for the $GI/M/1$ queue with phase-type vacations, *Journal of the Korean Statistical Society* **1**(40): 55–62.
- Korolyuk, V. (1975). *Boundary-Value Problems for Complicated Poisson Processes*, Naukova Dumka, Kiev, (in Russian).
- Korolyuk, V., Bratiichuk, M. and Pirdzhanov, B. (1987). *Boundary-Value Problems for Random Walks*, Ylym, Ashkhabad, (in Russian).
- Lee, T.T. (1984). $M/G/1/N$ queue with vacation time and exhaustive service discipline, *Operations Research* **4**(32): 774–784.
- Lillo, R.E. (2000). Optimal operating policy for an $M/G/1$ exhaustive server-vacation model, *Methodology and Computing in Applied Probability* **2**(2): 153–167.
- Mancuso, V. and Alouf, S. (2012). Analysis of power saving with continuous connectivity, *Computer Networks* **56**(10): 2481–2493.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edn., Springer-Verlag, New York, NY.
- Niu, Z., Shu, T. and Takahashi, Y. (2003). A vacation queue with setup and close-down times and batch Markovian arrival processes, *Performance Evaluation* **3**(54): 225–248.
- Niu, Z. and Takahashi, Y. (1999). A finite-capacity queue with exhaustive vacation/close-down/setup times and Markovian arrival processes, *Queueing Systems* **31**(1/2): 1–23.
- Nowak, A. and Woźniak, M. (2008a). Algorithm for optimization of the active module by the use of genetic algorithm, *Acta Mechanica Slovaca* **3**(C): 307–316.
- Nowak, A. and Woźniak, M. (2008b). Multiresolution derives analysis of module mechatronical systems, *Mechanika* **6**(74): 45–51.
- Nowak, A. and Woźniak, M. (2009). Optimization of the active vibroisolation system for operator’s cabin with the hydropneumatical element, *Transactions of the Universities of Košice* **1**(3): 113–116.
- Piórkowski, A. and Werewka, J. (2010). Minimization of the total completion time for asynchronous transmission in a packet data-transmission system, *International Journal of Applied Mathematics and Computer Science* **20**(2): 391–400, DOI: 10.2478/v10006-010-0029-z.
- Rutkowski, L. (2004). Adaptive probabilistic neural networks for pattern classification in time-varying environment, *IEEE Transactions on Neural Networks* **15**(4): 811–827.
- Rutkowski, L., Przybył, A. and Cpałka, K. (2012). Novel online speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation, *IEEE Transactions on Industrial Electronics* **59**(2): 1238–1247.
- Schaefer, R., Byrski, A. and Smółka, M. (2012). The island model as a Markov dynamic system, *International Journal of Applied Mathematics and Computer Science* **22**(4): 971–984, DOI: 10.2478/v10006-012-0072-z.
- Takagi, H. (1993). *Queueing Analysis*, North-Holland, Amsterdam.
- Takagi, H. (1994). $M/G/1/N$ queues with server vacations and exhaustive service, *Operations Research* **5**(42): 926–939.
- Teghem, J. (1986). Control of the service process in a queueing system, *European Journal of Operational Research* **1**(23): 141–158.
- Teghem, J. (1987). Optimal control of a removable server in an $M/G/1$ queue with finite capacity, *European Journal of Operational Research* **1**(31): 358–367.
- Wieczorek, R. (2010). Markov chain model of phytoplankton dynamics, *International Journal of Applied Mathematics and Computer Science* **20**(4): 763–771, DOI: 10.2478/v10006-010-0058-7.
- Woźniak, M. (2013). On applying cuckoo search algorithm to positioning $GI/M/1/N$ finite-buffer queue with a single vacation policy, *Proceedings of the 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, Mexico City, Mexico*, pp. 59–64.

Woźniak, M., Kempa, W.M., Gabryel, M., Nowicki, R.K. and Shao, Z. (2014). On applying evolutionary computation methods to optimization of vacation cycle costs in finite-buffer queue, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada (Eds.), *13th International Conference, ICAISC 2014, Zakopane, Poland, June 1–5, 2014, Proceedings, Part I*, Lecture Notes in Artificial Intelligence, Vol. 8467, Springer International Publishing Switzerland, Cham, pp. 480–491.

Woźniak, M., Marszałek, Z., Gabryel, M. and Nowicki, R.K. (2013). Modified merge sort algorithm for large scale data sets, in L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh and J.M. Zurada (Eds.), *12th International Conference, ICAISC 2013, Zakopane, Poland, June 9–13, 2013, Proceedings, Part II*, Lecture Notes in Artificial Intelligence, Vol. 7895, Springer-Verlag, Berlin/Heidelberg, pp. 612–622.



Marcin Woźniak received his M.Sc. degree in mathematics in 2007 and his Ph.D. degree in computer science in 2012. He is an assistant in the Institute of Mathematics at the Silesian University of Technology in Gliwice, Poland. His research interests include computational intelligence, evolutionary computation methods, computer algorithms, system modeling, as well as various aspects of applied mathematics and operations research.



Wojciech M. Kempa received his M.Sc. degree in mathematics in 1998, his Ph.D. degree in mathematics in 2003 and his D.Sc. degree in computer science in 2014. He is an assistant professor in the Institute of Mathematics at the Silesian University of Technology in Gliwice, Poland. He is a member of the Polish Mathematical Society and the American Mathematical Society. His research interests include queueing systems, telecommunications and computer networks, stochastic modeling, and various aspects of applied probability, statistics and operations research.



machine learning.

Marcin Gabryel received his M.Sc. degree in 2002 and his Ph.D. degree in 2007, both in computer science. He is an assistant professor at the Institute of Computational Intelligence of the Czestochowa University of Technology. He co-organizes the International Conference on *Artificial Intelligence and Soft Computing*. His research interests include developing new methods in computational intelligence, evolutionary design of fuzzy systems and ensembling methods in machine learning.



Robert K. Nowicki received his M.Sc. degree in electrical engineering in 1997, and his Ph.D. and D.Sc. degrees in computer science in 2000 and 2010, respectively. He is an associate professor at the Institute of Computational Intelligence of the Czestochowa University of Technology. His research interests include fuzzy systems, artificial neural networks, genetic algorithms, rough sets and hybrid methods.

Received: 16 October 2013

Revised: 31 March 2014