International Journal of
Mathematical Modelling & Computations
Vol. 09, No. 02, Spring 2019, 113- 122

Bat Algorithm for Optimal Service Parameters in an Impatient Customer N-Policy Vacation Queue

P. Vijaya Laxmi*

Department of Applied Mathematics, Andhra University, Visakhapatnam 530 003, India.

Abstract. In this paper, a meta-heuristic method, the Bat Algorithm, based on the echolocation behavior of bats is used to determine the optimum service rate of a queue problem. A finite buffer $M/M/1$ queue with N policy, multiple working vacations and Bernoulli schedule vacation interruption is considered. Under the two customers' impatient situations, balking and reneging, the queue is studied using the matrix geometric method. Simulations show that the proposed algorithm seems much superior to other algorithms.

Received: 05 March 2019, Revised: 20 April 2019, Accepted: 26 June 2019.

Keywords: Bat algorithm; Working vacations; Balking; Reneging; Queue.

AMS Subject Classification: 60K25, 68M20, 90B22.

Index to information contained in this paper

1. Introduction

Many queueing situations arise in real life wherein the customers are discouraged by a long queue. due to which, they either decide not to join the queue (known as balking) or depart after joining the queue without receiving service (reneging). These impatient acts lead to potential losses in revenue to service providers. For literatures related to impatience queue, see Haight [3], Ancker and Gafarian [2], Abou-El-Ata and Shawky [1], etc.

*Corresponding author. Email: vijaya_iit2003@yahoo.co.in; vijayalaxmiau@gmail.com

In working vacation model, the server after serving the queue exhaustively, goes for a vacation, and serves during the vacation period generally with a slower rate, see Servi and Finn [5], Wu and Takagi [9], Yue et al. [11]. Under the Bernoulli schedule vacation interruption (BS VI), the server is allowed to interrupt the vacation with probability q, if there are customers in the queue at a service completion epoch during WV, or continues the vacation with probability $1-q$. Zhang and Shi [12] first studied an $M/M/1$ queue with BS VI. Li et al. [6] studied a $GI/M/1$ queue with start-up period, SWV and BS VI using embedded Markov chain technique.

Queueing models with N-policy consider the most common issue of controlling services and to reduce the switching and set-up costs, Zhang and Xu [13]. Recently, Vijaya Laxmi and Jyothsna [7, 8] have analyzed an impatient customer $M/M/1$ queue with BS VI and VI queue with N policy, respectively.

Motivated by such situations, we study an N policy BS VI queue with balking and reneging that find practical applications in real life situations like hospitals, manufacturing processes, data transmission protocols, etc. We intend to propose a meta-heuristic method, namely, the Bat Algorithm (BA), based on the echolocation behavior of bats. The capability of echolocation of microbats is fascinating as these bats can find their prey and discriminate different types of insects even in complete darkness. Such echolocation behavior of microbats can be formulated in such a way that it can be associated with the objective function to be optimized, and this make it possible to formulate new optimization algorithms. In the rest of this paper, we will first outline the basic formulation of queue model and then discuss the implementation of BA through a cost function.

2. Model description

Let us consider an $M/M/1/K$ queue with balking, reneging and working vacations. The capacity of the system is finite K. The customers arrive one at a time according to a Poisson process with rate λ. Let b_n be the joining probability of an arriving customer and $1-b_n$ is the probability of balking, when there are n customers ahead of him. Furthermore, we assume that

$$0 \leqslant b_{n+1} \leqslant b_n < 1,\ 1 \leqslant n \leqslant K-1,\ b_0 = 1,\ b_K = 0.$$

The server commences MWV of random length when the system empties. The vacation times are exponentially distributed with rate ϕ. The service times during regular busy period as well as during working vacation period are independent and follow exponential distribution with parameters μ and η, respectively. At a service completion epoch during WV, if there are N or more customers in the queue, the server switches to regular busy period with probability q; otherwise, he remains in WV with probability $1-q$, i.e, a BS VI occurs depending on the number in the queue at a service completion instant during WV. The customers are served with FCFS service rule.

During vacation, each customer in the queue waits a certain length of time for service to begin before they leave the queue. This time T is assumed to follow exponential distribution with mean $1/\alpha$. The average reneging rate is given by $r(n) = (n-1)\alpha,\ 1 \leqslant n \leqslant K,$ and $r(n) = 0,\ n > K$. A brief diagrammatic representation of the model is shown in Figure 1.

3. Matrix-geometric solution

At steady-state, let $P_{0,n},\ 0 \leq n \leq K,\ \ (P_{1,n},\ 1 \leq n \leq K)$ be the probability that there are n customers in the system when the server is in working vacation

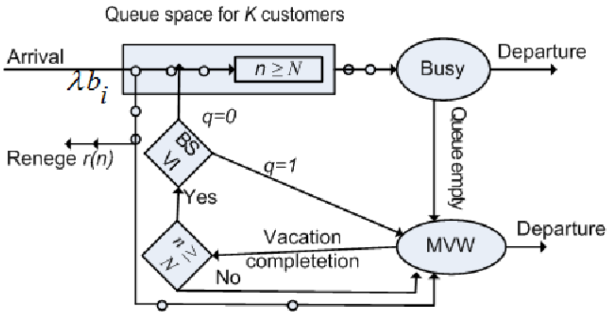

Figure 1. Block diagram of the queue model.

(regular busy period). We use the matrix-geometric method to obtain the steady-state probabilities. The transition rate matrix Q of the Markov chain has the block-trigonal form:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B_0} & \mathbf{C_0} & & & & & & & \\ \mathbf{A_1} & \mathbf{B_1} & \mathbf{C_1} & & & & & & \\ & \mathbf{A_2} & \mathbf{B_2} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots & \vdots \\ & & & \mathbf{A_N} & \mathbf{B_N} & \mathbf{C_N} & & & \\ & & & & \mathbf{A_{N+1}} & \mathbf{B_{N+1}} & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\ & & & & & & \mathbf{A_{K-1}} & \mathbf{B_{K-1}} & \mathbf{C_{K-1}} \\ & & & & & & & \mathbf{A_K} & \mathbf{B_K} \end{bmatrix}.$$

The rate matrix Q of this state process is similar to the quasi birth and death type and its elements are given below:

$$\begin{aligned} \mathbf{B_0} &= -\lambda b_0,\ \mathbf{C_0} = \lambda \mathbf{b_0},\ \mathbf{A_1} = (\eta+\alpha,\mu)^{\mathbf{T}} \\ \mathbf{B_i} &= \begin{bmatrix} -(\lambda b_i+\eta+i\alpha) & 0 \\ 0 & -(\lambda b_i+\mu) \end{bmatrix},\ i=1,2,\dots N-1, \\ &= \begin{bmatrix} -(\lambda b_i+\phi+\eta+i\alpha) & \phi \\ 0 & -(\lambda b_i+\mu) \end{bmatrix},\ i=N,N+1,\dots K, \\ \mathbf{C_i} &= \begin{bmatrix} \lambda b_i & 0 \\ 0 & \lambda b_i \end{bmatrix},\ i=1,2,\dots K-1, \\ \mathbf{A_i} &= \begin{bmatrix} i\alpha+\eta & 0 \\ 0 & \mu \end{bmatrix},\ i=2,3,\dots N, \\ &= \begin{bmatrix} i\alpha+\bar{q}\eta & q\eta \\ 0 & \mu \end{bmatrix},\ i=N+1,N+2,\dots,K, \end{aligned}$$

Let $\mathbf{P}$ denote the steady-state probability vector of $\mathbf{Q}$. By partitioning the vector $\mathbf{P}$ as $\mathbf{P} = \{\mathbf{P_0},\mathbf{P_1},\ \dots,\mathbf{P_N},\mathbf{P_{N+1}},\dots,\mathbf{P_K}\}$ where $\mathbf{P_0} = P_{0,0}$, $\mathbf{P_i} = (P_{0,i},P_{1,i})$ for $1 \leqslant i \leqslant K$. The steady-state equations $\mathbf{PQ} = \mathbf{0}$ are given by

$$\mathbf{P_0B_0} + \mathbf{P_1A_1} = 0, \tag{1}$$

$$\mathbf{P_{i-1}C_{i-1}} + \mathbf{P_iB_i} + \mathbf{P_{i+1}A_{i+1}} = 0,\ 1 \leqslant i \leqslant K-1, \tag{2}$$

$$\mathbf{P_{K-1}C_{K-1}} + \mathbf{P_KB_K} = 0, \tag{3}$$

From equations (1), (2) and (3), we get the recursive relation

$$\mathbf{P_i} = -\mathbf{P_{i+1}}\theta_{\mathbf{i}}, \quad \mathbf{i} = \mathbf{0}, \mathbf{1}, \ldots, \mathbf{K}-\mathbf{1}, \tag{4}$$

$$\mathbf{P_K}(\mathbf{B_K} - \theta_{\mathbf{K-1}}\mathbf{C_{K-1}}) = \mathbf{0}, \tag{5}$$

with

$$\theta_{\mathbf{i}} = \mathbf{A_{i+1}}(\mathbf{B_i} - \theta_{\mathbf{i-1}}\mathbf{C_{i-1}})^{-\mathbf{1}}, \mathbf{1} \leqslant \mathbf{i} \leqslant \mathbf{K}-\mathbf{1}, \quad \theta_{\mathbf{0}} = \mathbf{A_1}\mathbf{B_0}^{-\mathbf{1}}. \tag{6}$$

Equation (5) determines $\mathbf{P_K}$ up to a multiplicative constant. The other equation (4) determines $\mathbf{P_0}$, $\mathbf{P_1}, \ldots, \mathbf{P_{K-1}}$, up to the same constant, which is uniquely determined by the normalizing constant $\sum_{i=0}^{K} \mathbf{P_i}\, \mathbf{e} = \mathbf{1}$, where $\mathbf{e}$ is a column vector with each component equal to one. We develop a computer program to evaluate $\mathbf{P_i}$ and thereby we get $P_{0,i}$, $0 \leqslant i \leqslant K$ and $P_{1,i}$, $1 \leqslant i \leqslant K$.

4. Performance measures

Once the steady-state probabilities are known, one can obtain the various performance measures like expected queue length (L_q), expected number of customers in the system (L_s), probability that the server is busy with regular service (P_b), probability that the server is in working vacation (P_{wv}), probability that the server is idle (P_{id}), etc. They are given as

$$L_q = \sum_{n=1}^{N}(n-1)(P_{0,n} + P_{1,n}); \quad L_s = \sum_{n=1}^{N} n(P_{0,n} + P_{1,n}),$$

$$P_b = \sum_{n=1}^{N} P_{1,n} \quad ; \quad P_{wv} = \sum_{n=0}^{N} P_{0,n} \quad ; \quad P_{id} = P_{1,0} \ .$$

We can obtain the average balking rate (BR), the average reneging rate (RR) and the average rate of loosing a customer (LR) because of impatience. They are given by

$$BR = \sum_{i=0}^{1}\sum_{n=1}^{N}[\lambda(1-b_n)]P_{i,n}; \quad RR = \sum_{i=0}^{1}\sum_{n=1}^{N}(n-1)\alpha P_{i,n}; \quad LR = BR + RR.$$

5. Cost model

In practice, queueing managers are interested in minimizing operating cost or maximizing business profits. Under a given cost/revenue structure, we use the performance measures to search for the cost minimization with respect to the service rate μ by establishing an appropriate cost function. Following cost elements associated with various activities are considered:

- C_μ and C_η are the service costs every unit time during the normal working level and vacation period, respectively,
- C_{lq} represent the unit time cost of every waiting customer,
- C_{lr} be the cost per unit time when a customer balks or reneges.

Using these cost elements, we can establish the expected net cost function $F(\mu)$ as:

$$F(\mu) = C_\mu \, \mu/N + C_\eta \eta + C_{lq} L_q + C_{lr} \, LR.$$

The first two costs are incurred by the server, the third one by the customer's waiting and the fourth one by the customer loss. The cost minimization problem can be mathematically describes as an unconstrained problem as follows:

$$F(\mu^*) = \min_{\mu} F(\mu).$$

Bat algorithm

Modern optimization techniques are intuitively nature-inspired based on swarm intelligence and biological behavior of animals like ants, cuckoos, etc. Meta-heuristics such as genetic algorithm, particle swarm optimization, ant colony optimization are convenient algorithms. However, they have the drawback in dealing with the complex multi-modal optimization problems. The bat algorithm is an intelligence optimization algorithm inspired by the echolocation behavior of bats [10]. The following three idealized rules are assumed:

- All bats use echolocation to sense distance, and they also know the difference between food/prey and background barriers in some magical way.
- Bats fly randomly with velocity v_i at position x_i with a frequency f_{min}, varying wavelength Λ and loudness A_0 in search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.
- Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

Each bat is associated with a velocity v_i^t and a location x_i^t at iteration t, in a d dimensional search for solution space. Among all the bats, there exists a current best solution x^*. Therefore, the above three rules can be translated into updating equations for x_i^t and velocities v_i^t as

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \tag{7}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*) f_i, \tag{8}$$

$$x_i^t = x_i^{t-1} + v_i^t, \tag{9}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. We can either use wavelengths or frequencies for implementation, we will use $f_{min} = 0$ and f_{max} depending on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $[f_{min}, f_{max}]$. For this reason, bat algorithm can be considered as a frequency-tuning algorithm to provide a balanced combination of exploration and exploitation. The loudness and pulse emission rates essentially provide a mechanism for automatic control and auto zooming into the region with promising solutions.

Variations of loudness and pulse rates

In order to provide an effective mechanism to control the exploration and exploitation and switch to exploitation stage when necessary, we have to vary the loudness A_i and the pulse emission rate r_i during each iteration. Since the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience, between A_{min} and A_{max}. The assumption $A_{min} = 0$ means that a bat has just found the prey and temporarily stop emitting any sound. With these assumptions, we have

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0(1 - e^{-\gamma t}), \tag{10}$$

where α and γ are constants. For any $0 < \alpha < 1$ and $\gamma > 0$, we have

$$A_i^t \to 0, r_i^t \to r_i^0, \texttt{as } t \to \infty.$$

In the simplest case, we can use $\alpha = \gamma$, and we have used $\alpha = \gamma = 0.5$ `to` 0.9 in our computations. Note that if we replace the variations of the frequency f_i by a

Pseudo Code of the Bat Algorithm (BA)

Set-up the Objective function $f(x)$
Initialize the bat population $x_i, i = 1, 2, \cdots, n$ and v_i
Define pulse frequency f_i at x_i
Initialize pulse rates r_i and the loudness A_i
while ($t <$ Max number of iterations)
  Generate new solutions by adjusting frequency, and
  update velocities and locations/solutions [equations (7) to (9)]
  if ($rand > r_i$)
    Select a solution among the best solutions
    Generate a local solution around the selected best solution
  end if
  Generate a new solution by flying randomly
  if ($rand < A_i$ and $f(x_i) < f(x^*)$)
    Accept the new solutions
    Increase r_i and reduce A_i
  end if
  Rank the bats and find the current best x^*
end while

---

random parameter and setting $A_i = 0$ and $r_i = 1$, the bat algorithm becomes the standard Particle Swarm Optimization (PSO).

6. Numerical results

Numerous computations have been performed to study the parameter effect on the queueing model. However, a few results have been presented here in the form of tables and graphs. The parameter are taken as $K = 20,\ N = 3,\ \lambda = 2.2, \mu = 2.75, \eta = 0.5,\ q = 0.5$, unless otherwise specified. The balking function is taken as $b_n = 1 - (n/K^2),\ 1 \leq n \leq K - 1,\ b_0 = 1,\ b_K = 0$. For the cost analysis, we have taken $C_{lq} = 25; C_\mu = 18; C_\eta = 15; C_{lr} = 10$.

Figure 2 shows the graph of N versus average rate of loosing customers (LR) which indicated that the lower α has no effect whereas as α increases there is a definite increase in LR as is expected in practice. Figures 3, 4 and 5 depict variation of L_q with regard to η for different α and q values. From Figures 2 and 3, we observe that as for all values of ϕ and q, L_q decreases as η increases and becomes equal at

Table 1. Optimum cost for different N and λ for $\alpha = 0.3,\ \eta = 0.4,\ \phi = 1.2,\ K = 10$.

(N, λ)	$(\mu^*,\ F^*(\mu))$	L_s	BR	RR	LR
	$q = 1.0$ (Complete priority for VI)				
(3, 1.2)	(2.33546, 36.4286)	1.23123	0.0147993	0.174298	0.189098
(3, 1.5)	(2.78657, 43.6183)	1.43417	0.0216211	0.2216	0.243221
(3, 2.5)	(4.12742, 66.8982)	2.06384	0.0541857	0.381341	0.435527
(5, 1.2)	(2.83802, 30.2882)	1.1119	0.0133588	0.149333	0.162692
(5, 1.5)	(3.36545, 36.3141)	1.3048	0.0196506	0.192878	0.212529
(5, 2.5)	(4.90805, 56.1606)	1.90825	0.0499259	0.342775	0.392701
	$q = 0.5$ (Partial priority for VI)				
(3, 1.2)	(2.32211, 36.8516)	1.25606	0.0150998	0.179656	0.194756
(3, 1.5)	(2.76658, 44.2714)	1.46968	0.0221686	0.229824	0.251992
(5, 1.2)	(2.82225, 30.7459)	1.13801	0.0136743	0.154811	0.168485
(5, 1.5)	(3.34148, 37.0193)	1.34201	0.0202218	0.201298	0.22152

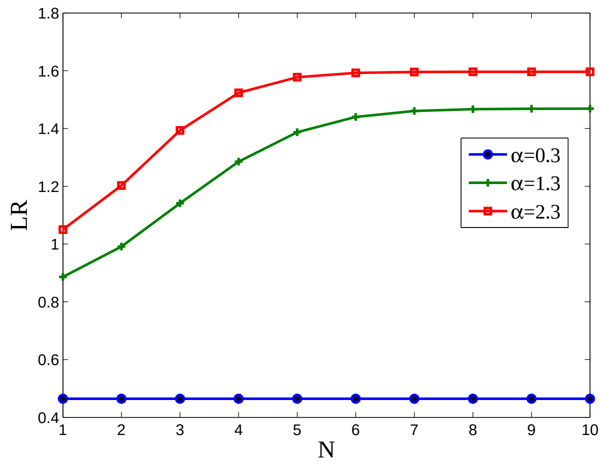

Figure 2. N versus LR for different α.

$\eta = \mu = 2.75$. Upon increasing η further a marginal reverse nature in the graph is observed. In Figure 4, the lower the reneging rate, the larger is the queue length. Thus a balancing factor of reneging is to maintained to keep the queue length at an optimum level as shown in Figure 5.

Table 1 shows the optimum $\mu^*, F(\mu^*)$ for different values of arrival rate λ, N and vacation interruption parameter q. There is an obvious increase in μ^* and $F(\mu^*)$ with the increase of λ and N. However, the system incurs lesser cost and services whenever there is a complete tendency to switch over to regular service ($q = 1$). This is intuitively true, as services are rendered faster during regular service periods.

Figure 7 shows the generations of BA versus cost and its faster convergence. Figure 8 compares the number of iterations versus the μ value for different buffer content K and the comparison is done with the standard optimization algorithm quadratic fit search method (QFSM), see Rardin [4].

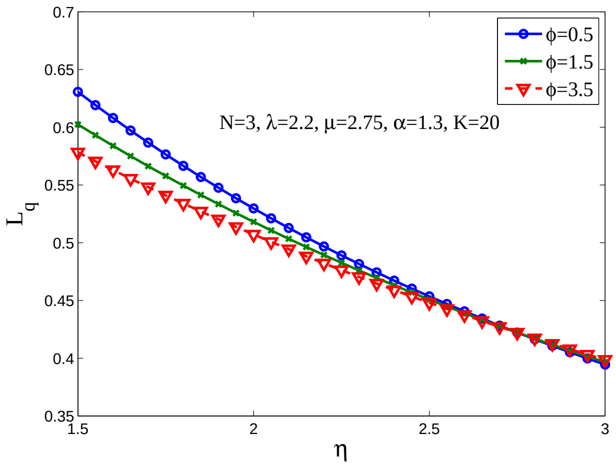

Figure 3. η versus L_q for various ϕ.

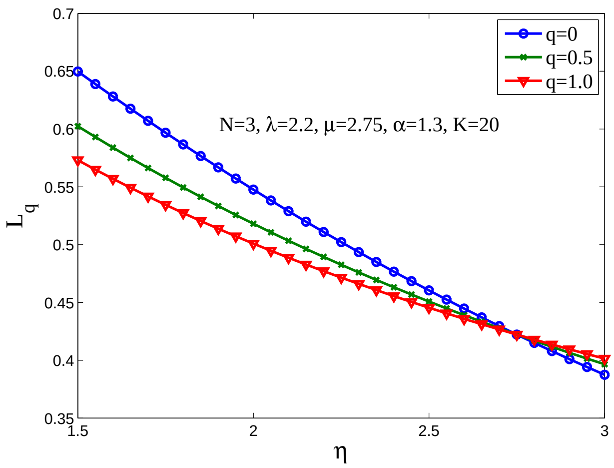

Figure 4. η versus L_q for different q.

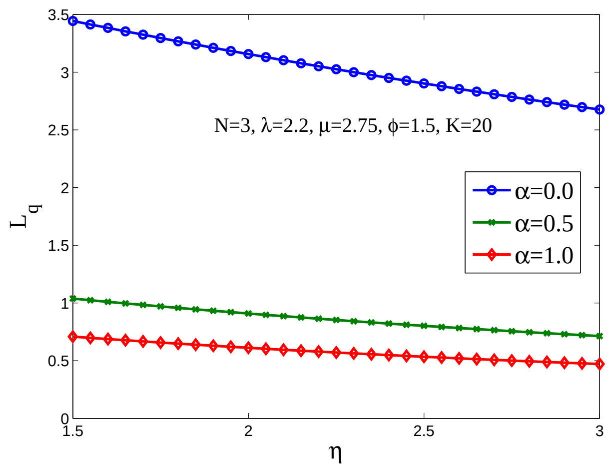

Figure 5. η versus L_q for various α.

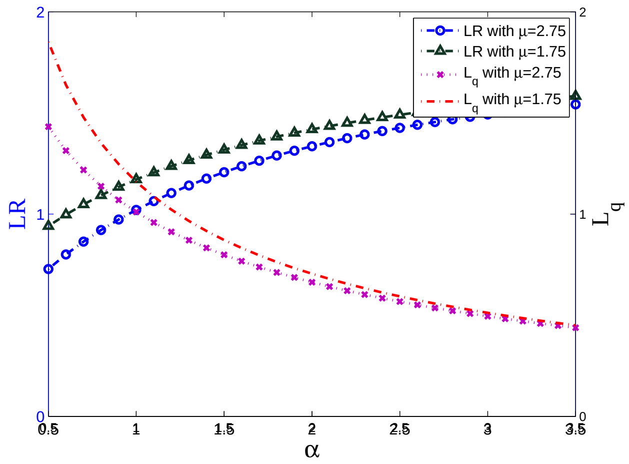

Figure 6. Impact of α on LR and L_q.

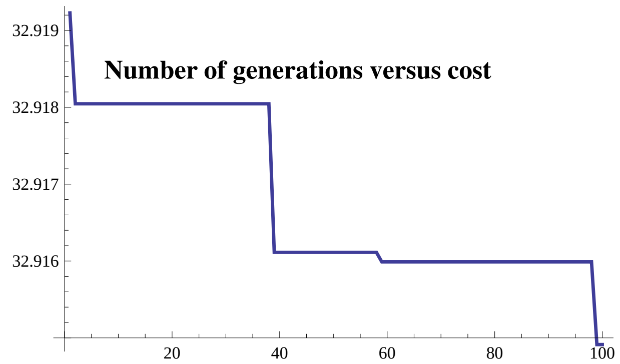

Figure 7. Generations versus cost.

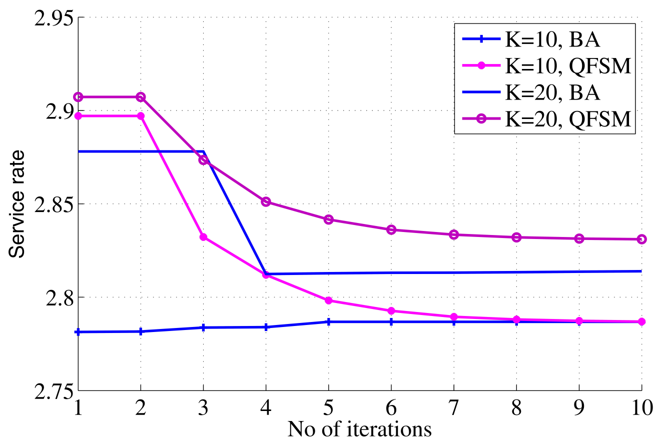

Figure 8. Number of iterations versus service rate μ.

7. Conclusions

In this paper, we have carried out an analysis of a finite buffer $M/M/1$ queue with balking, reneging and multiple working vacations with Bernoulli-schedule vacation interruptions. Some numerical results are presented in the form of tables and graphs and cost minimization problem has been discussed using bat algorithm which is a meta-heuristic optimization technique of nature-inspired swarm intelligence and biological behavior of animals like ants, cuckoos, etc. The bat algorithm is found to be much efficient and has more faster convergence compared to other optimization techniques.

References

[1] M. O. Abou-El-Ata and A. I. Shawky, The single-server Markovian overflow queue with balking, reneging and an additional server for longer queues, Microelectronics Reliability, **32 (10)** (1992) 1389-1394.

[2] C. J. Ancker and A. V. Gafarian, Some queueing problems with balking and reneging. I., Operations Research, **11 (1)** (1963) 88-100.

[3] F. A. Haight, Queueing with reneging, Metrika, **2** (1959) 186-197.

[4] R. L. Rardin, Optimization in Operations Research, Prentice Hall, New Jersy, (2008).

[5] L. D. Servi and S. G. Finn, M/M/1 queues with working vacations (M/M/1/WV), Performance Evaluation, **50 (1)** (2002) 41-52.

[6] L. Tao, Z. Liu and Z. Wang, The GI/M/1 queue with start-up period and single working vacation and Bernoulli vacation interruption, Applied Mathematics and Computation, **218 (8)** (2011) 4401–4413.

[7] P. Vijaya Laxmi, V. Goswami and K. Jyothsna, Optimization of balking and reneging queue with vacation interruption under N-policy, Journal of Optimization, **2013** (2013), Article ID 683708, doi: 10.1155/2013/683708.

[8] P. Vijaya Laxmi and K. Jyothsna, Impatient customer queue with Bernoulli schedule vacation interruption, Computers & Operations Research, **56** (2015) 1-7.

[9] D. Wu and H. Takagi, M/G/1 queue with multiple working vacations, Performance Evaluation, **63 (7)** (2006) 654-681.

[10] X. S. Yang A new metaheuristic bat-inspired algorithm, in Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) (Eds. J. R. Gonzalez et al.), **SCI 284** (2010) 65-74.

[11] D. Yue, Y. Zhang and W. Yue, Optimal performance analysis of an M/M/1/N queue system with balking, reneging and server vacation, International Journal of Pure and Applied Mathematics, **28 (1)** (2006) 101-115.

[12] H. Zhang and D. Shi, The M/M/1 queue with Bernoulli-schedule-controlled vacation and vacation interruption, International Journal of Information and Management Sciences, **20 (4)** (2008) 579-587.

[13] Z. Zhang and X. Xu, Analysis for the M/M/1 queue with multiple working vacation and N-policy, International Journal of Information and Management Sciences, **19 (3)** (2008) 495-506.