

Firefly algorithm in optimization of queueing systems

J. KWIECIEN^{*} and B. FILIPOWICZ

AGH University of Science and Technology, Department of Automatics, 30 Mickiewicza Ave., 30-059 Krakow, Poland

Abstract. Queueing theory provides methods for analysis of complex service systems in computer systems, communications, transportation networks and manufacturing. It incorporates Markovian systems with exponential service times and a Poisson arrival process. Two queueing systems with losses are also briefly characterized. The article describes firefly algorithm, which is successfully used for optimization of these queueing systems. The results of experiments performed for selected queueing systems have been also presented.

Key words: queueing systems, firefly algorithm.

1. Introduction

Nature-inspired metaheuristics are currently among the most powerful tools for optimization of many NP-hard combinatorial problems. These methods are based on existing mechanisms of a biological phenomenon of nature. The natural systems are the ones of the most interesting inspiration for designing new techniques dedicated to solve many optimization problems. The ant systems, particle swarm optimization and bee algorithms are the methods inspired from observing nature. These algorithms use the behavior of swarm intelligence. So they are based on alive insects or simple interactions among individual entities [1, 2]. These metaheuristics are becoming more and more popular.

The firefly algorithm (FA), proposed by Xin-She Yang at Cambridge University, is a novel metaheuristic, which is inspired by the behavior of fireflies. Their population is estimated about two thousand firefly species. Most of them produce short and rhythmic flashes. Their flashing light generated by a process of bioluminescence may serve as an element of courtship rituals or warning signals [1–3].

The rest of the paper is organized as follows. In Sec. 2 we briefly describe the firefly algorithm. Section 3 presents two queueing systems with losses. Section 4 gives description of optimization problems in queueing systems and experimental results based on the firefly algorithm, which is used to solve selected optimization problems. Finally, Sec. 5 summarizes the conclusions.

2. Firefly algorithm

In the firefly algorithm, the objective function of a given optimization problem is based on differences in light intensity. It helps the fireflies to move towards brighter and more attractive locations in order to obtain optimal solutions. All fireflies are characterized by their light intensity associated with the objective function. Each firefly is changing its position iteratively. The firefly algorithm has three rules [1–3]:

- All fireflies are unisex, and they will move towards more attractive and brighter ones.
- The attractiveness of a firefly is proportional to its brightness which decreases as the distance from the other firefly increases. If there is not a more attractive firefly than a particular one, it will move randomly.
- The brightness of a firefly is determined by the value of the objective function. For maximization problems, the brightness is proportional to the value of the objective function.

Each firefly has its attractiveness β described by monotonically decreasing function of the distance r between two any fireflies [1]:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad m \geq 1, \quad (1)$$

where β_0 denotes the maximum attractiveness (at $r = 0$) and γ is the light absorption coefficient, which controls the decrease of the light intensity.

The distance between two fireflies i and j at positions x_i and x_j can be defined as follows [1]:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (2)$$

where $x_{i,k}$ is the k -th component of the spatial coordinate x_i of i -th firefly and d denotes the number of dimensions.

The movement of a firefly i is determined by the following form [1]:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \left(rand - \frac{1}{2} \right), \quad (3)$$

where the first term is the current position of a firefly i , the second term denotes a firefly's attractiveness and the last term is used for the random movement if there are not any brighter firefly (rand is a random number generator uniformly distributed in the range $< 0, 1 >$). For most cases $\alpha \in (0, 1)$, $\beta_0 = 1$. In practice the light absorption coefficient γ varies from 0.1 to 10. This parameter describes the variation of the attractiveness and its value is responsible for the speed of FA convergence [1].

*e-mail: kwiecien@agh.edu.pl

The firefly algorithm can be presented in the following pseudo-code form [1, 2]:

1. Initialize algorithm's parameters:
 - number of fireflies (n),
 - β_0, γ, α ,
 - maximum number of generations (iterations, MaxGen).

Define the objective function $f(x), x = (x_1, \dots, x_d)^T$.

Generate initial population of fireflies $x_i (i = 1, 2, \dots, n)$

//generate n initial solutions

Light intensity of firefly I_i at x_i is determined by value of objective function $f(x_i)$

2. While $k < \text{MaxGen} // (k = 1 : \text{MaxGen})$

For $i = 1:n // \text{all } n$ fireflies

For $j = 1:n$

If $(I_j > I_i)$ move firefly i towards firefly j in d -dimension according to Eq. (3); End if

Obtain attractiveness, which varies with distance r according to Eq. (1)

Find new solutions and update light intensity

End for j

End for i

Rank the fireflies and find the current best

End while

3. Find the firefly with the highest light intensity, visualization

The initial population of fireflies is generated in the following form:

$$x_i = LB + rand \cdot (UB - LB), \quad (4)$$

where LB and UB denote the lower and the upper bounds of i -th firefly.

After the evaluation of the initial population the firefly algorithm enters its main loop, which represents the maximum number of generations of the fireflies (iterations). For each generation the firefly with the maximum light intensity (the solution with the best value of objective function) is chosen as the potential optimal solution). The firefly algorithm simulates a parallel run strategy. The population of n fireflies generates n solutions.

Using the firefly algorithm we can solve many optimization problems [1–3]. Assume that example function, which describes the optimization problem is:

$$\max \rightarrow f(x, y) = \sin(x) + \cos(y), \quad (x, y) \in \langle -2\pi, 2\pi \rangle. \quad (5)$$

Figure 1 shows the surface of a given objective function. The results of firefly algorithm and the location of fireflies are shown in Figs. 2–4.

The firefly algorithm is very efficient metaheuristic in finding the global optima [1, 2]. In [3] the efficiency of the firefly algorithm was demonstrated for continuous constrained optimization task. The classic firefly algorithm works in the continuous space. It can be modified to seek optimum of discrete optimization landscapes. A binary version of the firefly algorithm was used in [4] for makespan optimization of permutation flowshop problem. After updating the firefly's posi-

tion according to Eq. (3), it is changed from real numbers by applying a probabilistic rule based on sigmoid function.

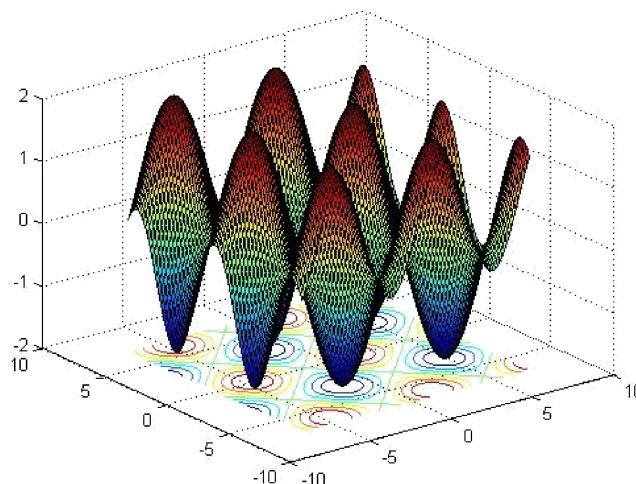


Fig. 1. Example function

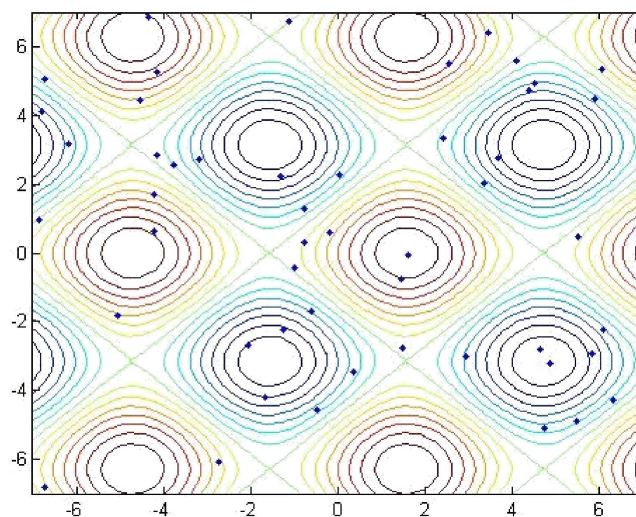


Fig. 2. The initial locations of 50 fireflies

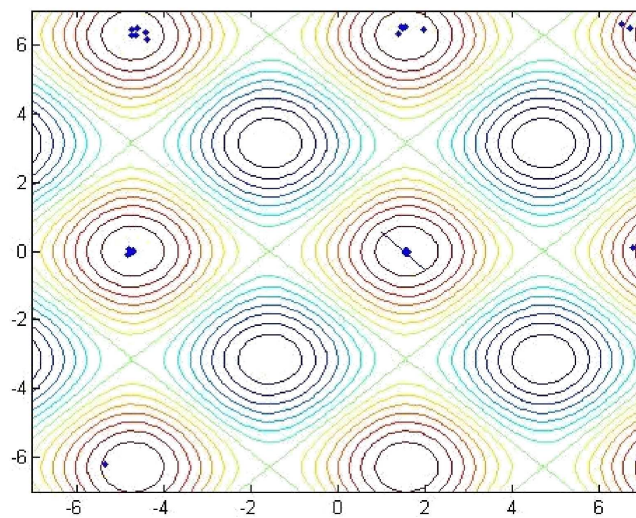


Fig. 3. The locations of fireflies after 50 iterations

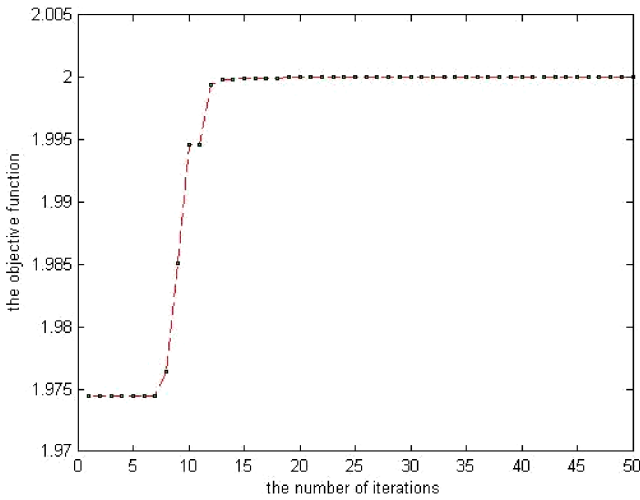


Fig. 4. The relationship of the objective function with iterationsm (generations)

3. Queueing systems

Here we consider the models with exponential service times, in which the arrival process is a Poisson one [5–7]. The mean arrival rate (per unit of time) at each system is denoted by λ and the parameter of the service time is denoted by μ . Traffic intensity ρ is the ratio of arrival λ to service rate μ . If the stationary probabilities are known, it is possible to derive all performance characteristics, for example the mean number of jobs in a system and in a queue, mean holding time or mean number of busy servers.

3.1. The M/M/m-m queueing system with losses. In this subsection we analyze the model with exponential interarrival times, exponential service times and m parallel identical servers. In this system arriving customer is served if at least one server is available. When all servers are occupied the newly arriving customer departs the queueing systems without being served. These customers are lost. The steady-state probability of the system being empty has the following form [6, 7]:

$$\pi_0 = \frac{1}{\sum_{k=0}^m \frac{\rho^k}{k!}} \quad (6)$$

The steady-state probability of k jobs in the system is as follows:

$$\pi_k = \frac{\frac{\rho^k}{k!}}{\sum_{k=0}^m \frac{\rho^k}{k!}} \quad (7)$$

The steady-state probability that the newly arriving customers are lost:

$$\pi_l = \pi_m = \frac{\frac{\rho^m}{m!}}{\sum_{k=0}^m \frac{\rho^k}{k!}} \quad (8)$$

3.2. The M/M/m/FIFO/m+N queueing system with finite capacity and impatient customers. In some situations, it may occur that the customers should be served before their respective deadlines. Therefore, to describe the problem we will consider the m -server queueing system with limited waiting room with FIFO queueing discipline and limited waiting time in the queue. In this system the maximum number of customers amounts to $m + N$, so there is a limited waiting room (N). Each job arriving to the system has its own maximal waiting time T_w . This time is assumed to be with an exponential distribution with parameter δ . If the time, which a job would have to wait for accessing a server exceeds T_w , then it departs from the system after time T_w . If newly arriving customers find $m + N$ customers in systems, they are lost. The state diagram for this system is shown in Fig. 5.

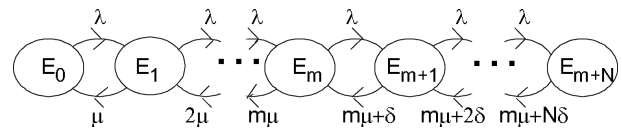


Fig. 5. State diagram for M/M/m/FIFO/m+N (E_i denotes state with i customers in system)

Let π_i denotes the probability that system is in state E_i . In the stationary case, we obtain the following set of linear algebraic equations derived according to the Chapman-Kolmogorov equations:

$$\begin{aligned} 0 &= -\lambda\pi_0 + \mu\pi_1 \\ 0 &= \lambda\pi_0 - (\lambda + \mu)\pi_1 + 2\mu\pi_2 \\ &\vdots \\ 0 &= \lambda\pi_{m-1} - (\lambda + m\mu)\pi_m + (m\mu + \delta)\pi_{m+1} \\ &\vdots \\ 0 &= \lambda\pi_{m+N-1} - (m\mu + N\delta)\pi_{m+N}. \end{aligned} \quad (9)$$

These probabilities also satisfy the normalization equation:

$$\sum_{i=0}^{m+N} \pi_i = 1. \quad (10)$$

Solving Eq. (9) we obtain the steady-state probabilities of this system.

The stationary probability of no jobs in a system is given by:

$$\pi_0 = \left[\sum_{k=0}^m \frac{\rho^k}{k!} + \frac{\rho^m}{m!} \sum_{r=1}^{m+N} \frac{\rho^r}{\prod_{n=1}^r \left(m + n \frac{\delta}{\mu} \right)} \right]^{-1}. \quad (11)$$

The probability that jobs will be lost because of exceeding the time limit is the following:

$$\pi_w = \left(\frac{\delta}{\lambda} \cdot \frac{\rho^m}{m!} \cdot \sum_{r=1}^{m+N} \frac{r\rho^r}{\prod_{n=1}^r \left(m + n \frac{\delta}{\mu} \right)} \right) \cdot \pi_0. \quad (12)$$

The steady-state probability that the newly arriving customers are lost because $m + N$ jobs exists in system has the following form:

$$\pi_{m+N} = \frac{\rho^{m+N} \pi_0}{m! \prod_{n=1}^N \left(m + n \frac{\delta}{\mu}\right)} \quad (13)$$

The probability that jobs will be lost is given by:

$$\pi_l = \pi_w \cdot \pi_{m+N} \quad (14)$$

4. Optimization of queueing systems by firefly algorithm

To formulate optimization problems we need to determine parameters that can be decision variables. Many of the system parameters may be the components of various optimization problems. There are some types of optimization, for example: minimization of cost for given throughput by changing of the service rate, maximization of throughput, minimization of the mean response time [5, 6]. We may consider the problem of optimizing costs controlled by e.g. the number of servers, the service rate, the capacity of the queue and the maximal waiting time.

In the case of M/M/m/-/m queueing systems with losses we seek the number of servers that maximizes the overall profits. Here we get the following objective function [6]:

$$\max \rightarrow f(m) = r \frac{\lambda}{\mu} \left(1 - \frac{\left(\frac{\lambda}{\mu}\right)^m}{m! \sum_{k=0}^m \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!}} \right) - cm, \quad (15)$$

where r denotes the cost related to jobs in system and c denotes the cost of server depreciation.

For M/M/m/FIFO/m+N queueing system with impatient customers we consider the maximization of profits subject to the number of servers and the number of waiting places. The objective function is given as follows:

$$\max \rightarrow f(m, N) = c_1 \lambda (1 - \pi_l) - c_2 (m + N), \quad (16)$$

where π_l is obtained using Eq. (14), c_1 is the profit on job service and c_2 denotes the cost of server depreciation.

The optimization of queueing systems introduced above is difficult. These are discrete optimization problems, and there is not any general method to solve these problems. These problems are known in literature, but according to our knowledge, they were not solved by these nature-inspired metaheuristics. It should be noted that there are studies on the use of the genetic algorithms to solve various problems of optimization of some queueing systems, for example [8].

To identify parameters of objective function we use the firefly algorithm. For this purpose, we specially prepared test

instances with known solutions in advance (obtained by using genetic algorithms and a complete review).

In order to solve the optimization problem we implemented it in Matlab under Windows XP operating system with help of files presented in [9]. The values of firefly algorithm parameters were: $\alpha = 0.1$, $\beta_0 = 0.1$, $\gamma = 1$. In the case of M/M/m/-/m system the number of fireflies is 20 and the maximum generation of fireflies is 50, so the total number of functions evaluation is 1000. The results of computational experiments for this system are shown in Table 1 and Fig. 6.

Table 1
The results of FA optimization for $m \in \langle 1, 100 \rangle$

M/M/m/-/m, $r = 2, c = 1$			
λ	μ	The optimum value of m	$f(m)$
10	2	5	2.1513
10	5	2	0.4
60	2	32	22.224
60	5	13	7.2824

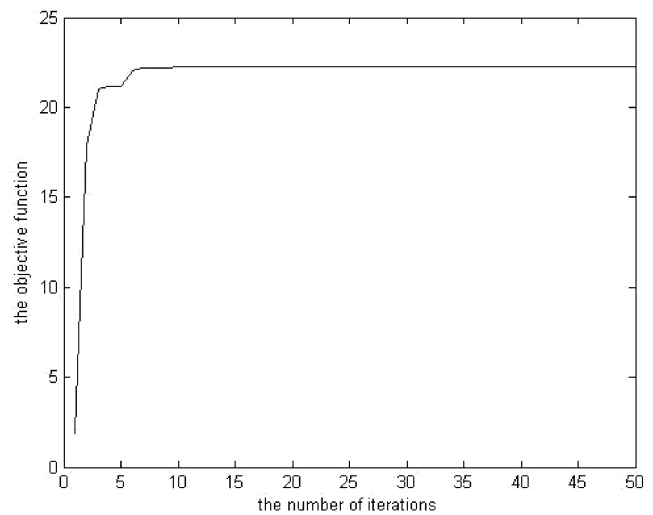


Fig. 6. The relationship of the objective function with iterations; (case: $\lambda = 60, \mu = 2$)

For optimization of M/M/m/FIFO/m+N queueing system with impatient customers we use 40 fireflies and 50 generations. Table 2 shows the experimental results for this system, if decision variable is only the number of servers.

Table 2
The results of FA optimization for $m \in \langle 1, 100 \rangle$

M/M/m/FIFO/m+N, $c_1 = c_2 = 5, \delta = 5$				
λ	μ	N	The optimum value of m	$f(m)$
100	2	10	4	407.046
200	2	10	3	927.657
100	20	10	4	426.099

Figures 7–9 show the changes of the objective function during solving optimization problems by FA.

Firefly algorithm in optimization of queueing systems

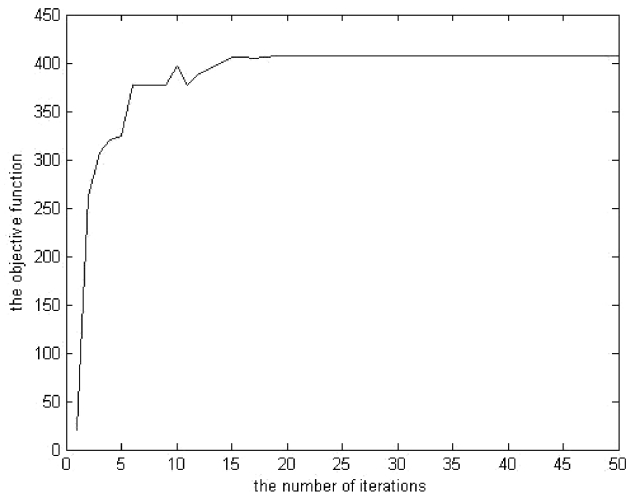


Fig. 7. The relationship of the objective function with iterations; (case: $\lambda = 100, \mu = 2, N = 10$)

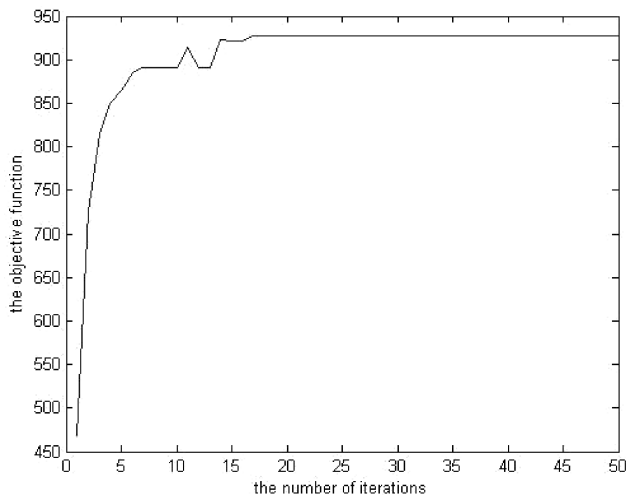


Fig. 8. The relationship of the objective function with iterations; (case: $\lambda = 200, \mu = 2, N = 10$)

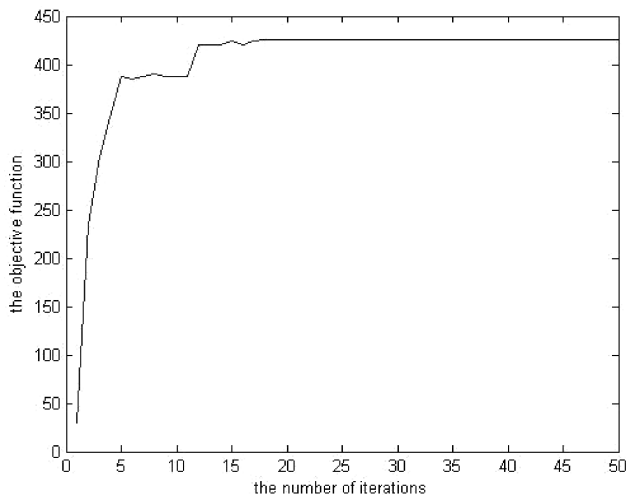


Fig. 9. The relationship of the objective function with iterations; (case: $\lambda = 100, \mu = 20, N = 10$)

Results of all experiments show that the firefly algorithm converges to the optimal solution very quickly. We can find

the maximum value of the objective function in about 18 iterations.

In order to test the efficiency of firefly algorithm we consider another case, that is the problem with two decision variables: the number of identical servers m and the number of waiting places N . Initial parameters of FA are identical as in the previous case. The results of conducted experiments are shown in Table 3 and Fig. 10–12. The results show that the firefly algorithm is very efficient. Using this algorithm we can reach the maximum value of the objective function and the optimum of unknown parameters within 20 generations.

Table 3
The results of FA optimization for $m \in \langle 1, 100 \rangle$ and $N \in \langle 1, 50 \rangle$

M/M/m/FIFO/m+N, $c_1 = c_2 = 5, \delta = 5$				
λ	μ	The optimum value of m	The optimum value of N	$f(m, N)$
100	2	1	1	485.043
200	2	1	1	987.266
100	20	1	1	481.529

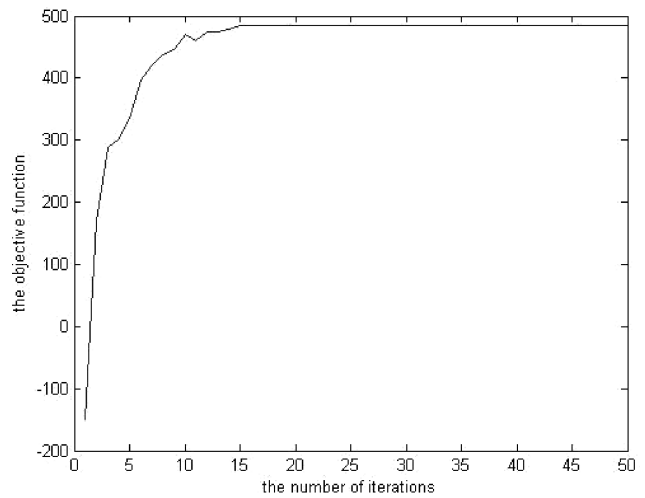


Fig. 10. The relationship of the objective function with iterations, $\lambda = 100, \mu = 2$

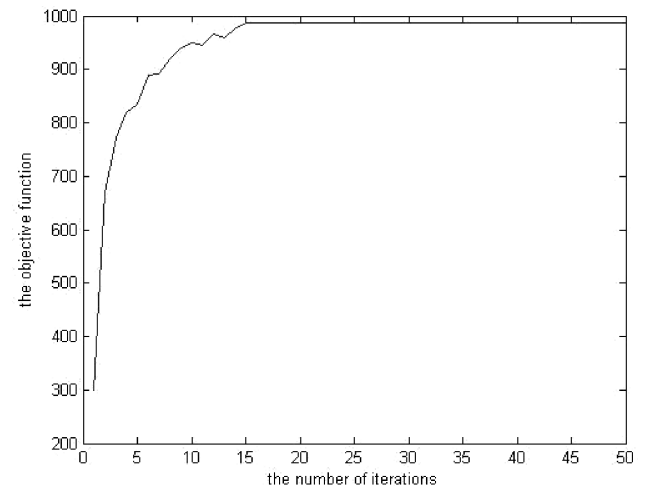


Fig. 11. The relationship of the objective function with iterations, $\lambda = 200, \mu = 2$

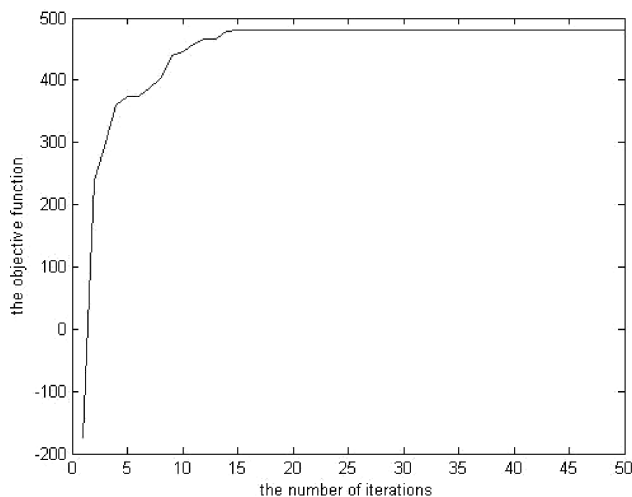


Fig. 12. The relationship of the objective function with iterations, $\lambda = 100$, $\mu = 20$

In all cases presented in this section we generate the initial position using $\text{round}(x_i)$.

5. Conclusions

The firefly algorithm is a very powerful technique used to solve the problems of queueing systems optimization. It is a simple method and easy to implement. In this paper we have tested this algorithm to the multiobjective maximization problem of cost function. The parameters of firefly algorithm such as the absorption coefficient, the population of fireflies and the number of iterations depend on the optimized problem. Our goal is to test the tools that in future research will be used for different optimization of queueing systems and queueing networks, including various types of blocking mechanisms [5, 10, 11]. Future work will focus on application of other metaheuristics such as the bee algorithm and particle swarm optimization in queueing optimization problems.

REFERENCES

- [1] X.S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, London, 2008.
- [2] X. S. Yang, "Firefly algorithms for multimodal optimization", *Stochastic Algorithms: Foundations and Applications, SAGA, Lecture Notes in Computer Sciences 5792*, 169–178 (2009).
- [3] S. Łukasik and S. Žak, "Firefly algorithm for continuous constrained optimization task", *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems LNCS 5796*, 97–106 (2009).
- [4] M.K. Sayadi, R. Ramezani, and N. Ghaffari-Nasab, "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems", *Int. J. Industrial Eng. Computations* 1, 1–10 (2010).
- [5] G. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi, *Queueing Networks and Markov Chains. Modeling and Performance Evaluation with Computer Science Applications*, John Wiley&Sons, London, 1998.
- [6] B. Filipowicz, *Modeling and Optimization of Queueing Systems. Volume 1, Markovian Systems*, T. Rudkowski Publishing House, Cracow, 1999, (in Polish).
- [7] B. Filipowicz and J. Kwiecień, "Queueing systems and networks: models and applications", *Bull. Pol. Ac.: Tech.* 56 (4), 379–390 (2008).
- [8] Ch.H. Lin and J.Ch. Ke, "Optimization analysis for an infinite capacity queueing system with multiple queue-dependent servers: genetic algorithms", *Int. J. Computer Mathematics* 88 (7), 1430–1442 (2011).
- [9] X.S. Yang, *Firefly Algorithm – Matlab files*, <http://www.mathworks.com/matlabcentral/fileexchange/29693-firefly-algorithm> (2011).
- [10] W. Oniszczyk, "Loss tandem networks with blocking – a semi-Markov approach", *Bull. Pol. Ac.: Tech.* 58 (4), 673–681 (2010).
- [11] S. Balsamo, V. de Nito Persone, and R. Onvural, *Analysis of Queueing Networks with Blocking*, Kluwer Academic Publishers, Boston, 2001.