Algorithm design for optimization of service requests for repair

L V Rossihina¹, A V Kalach¹², A Yu Akulov², T P Sysoeva³ and A S Kravchenko¹

¹Voronezh Institute of the Federal Penitentiary Service of Russia, 1-a Irkutsk Str., Voronezh, 394072, Russia
²Ural Institute of State Fire Service of EMERCOM of Russia, 22 Mira Str, Ekaterinburg, 620062, Russia
³Saint-Petersburg University of EMERCOM of Russia, Moscow avenue 149, Saint-Petersburg, 196105, Russia

E-mail: AVKalach@gmail.com

Abstract. Statement of the problem of service request for repair is formalized and algorithms of solution are presented. An algorithm is designed for the service of the greatest number of requests for repair within the established deadlines. An algorithm is proposed for constructing the optimal sequence of servicing repair requests. The optimal sequence is determined basing on the least total time of delays while completion of the requests service within the established deadlines. Usage of the optimization algorithms for the service of requests for repair in certain examples.

1. Introduction

Analysis of literature demonstrated that the problem of scheduling of the service with the account of the established deadlines and penalties for violation of certain deadlines was formalized in 60-th – 70-th of the last year [1, 2, 3]. Step-like penalty functions were studied in [4, 5]. The problems with minimization of the number of requests with a delayed service were considered in [6, 7, 8]. Generalization of these results for the case when a subset of requests is determined while each of them must be served is provided in [9]. Concept of the generalized intervals in priority was introduced in [10]. Sufficient conditions when some request from the given set will be served as the first (or the last) one are presented in [11].

2. Algorithm for the service of the greatest number of requests in the established deadlines

Let us consider the problem of optimization in servicing of requests for the repair in the established deadlines.

Statement of the problem. Let us have n requests for the repair which are served for the time of \( t_{ij} \), \( j = 1, n \); \( T_{ij} \) – is the established deadline of service for \( i \) request; \( Z_{ij} = \max(0; \bar{t}_{ij} - T_{ij}) \), \( \bar{t}_{ij} \) where is an actual time of service completion of the \( i \) request. The problem is to determine a sequence of servicing of the greatest number of requests for the repair during the deadline.

Let’s consider the description of the algorithm.

Step 1. The received requests for repair are numbered in the order of non-decreasing values of \( T_i \), \( j = 1, n \), i.e. the primary sequence of the requests for repair is formed \( \pi^{(0)} = (i_1, i_2, \ldots, i_n) \). If during the
requests servicing in sequence \( \pi^{(n)} \) all of the requests are served in time \( Z_i = 0 \) , then the sequence is an optimal one \( \pi \).

Step 2. Otherwise a sequence of requests ordered in non-decrease of their indexes is such that \( Z_{i_p} > 0 \) and \( Z_{i_j} = 0 \) for \( j < p \). Then a set of requests \( A = \{i_1, i_2, \ldots, i_p\} \) is formed and the request \( i_p \) is selected among them with the largest value of \( t_i \). This request is excluded from the new sequence of the requests for repair \( \pi^{(n-1)} = (i_1, i_2, i_q, 1) \).

Step 3. As a result of repeating step 2 for the finite number of times a new sequence will be obtained \( \pi^{(r)} = (i_1^*, i_2^*, \ldots, i_r^*) \), where \( Z_{i_j^*} = 0, j = 1, 2, r \) , as well as the optimal sequence for servicing of the requests for repair \( \pi^* = (\pi^{(r)}; \bar{\pi}) \) here \( \bar{\pi} \) is an arbitrary sequence of requests different from the requests in the set of \( (i_1^*, i_2^*, \ldots, i_r^*) \).

Example 1. It is required to determine the sequence of servicing the requests for repair when the greatest number of the repair activities is performed during the established deadline \( T_k \).

Let the repair activities numbered in the order of increase (non-decrease) of the established deadlines for completion of k-th request servicing \( T_k \) (table 1), \( t_k \) – is the time of the request servicing.

Table 1. Data on the requests for repair.

| k   | \( t_k \) hr. | \( T_k \) hr. |
|-----|--------------|--------------|
| 1   | 0.4          | 0.5          |
| 2   | 0.5          | 1            |
| 3   | 0.5          | 1            |
| 4   | 0.6          | 1.5          |
| 5   | 0.8          | 2            |
| 6   | 1            | 3            |
| 7   | 1.5          | 4            |

1. While servicing the requests for the repair \( \pi \) in the sequence of \( \pi^{(7)} = (1, 2, 3, 4, 5, 6, 7) \) one can obtain the actual time \( \bar{t}_k \) for completion of their servicing, that is equal to 0.4; 0.9; 1.4; 2; 2.8; 3.8 and 5.3, respectively. Then we calculate the values of \( Z_k = \max (0; \bar{t}_k - T_k) \). As a result, we obtain \( Z_1 = Z_2 = 0; Z_3 = 0.4; Z_4 = 0.5; Z_5 = 0.8; Z_6 = 0.8; Z_7 = 1.3 \). Since not of all \( Z_k = 0 \), then not of all the requests for repair can be served within the established deadline.

2. Let us form a set of \( A = \{1, 2, 3\} \) requests for the repair in in this set let us choose the request with the largest servicing time \( t_k \). In the considered example \( k = 3 \). Servicing the requests for repair in the sequence of \( \pi^{(3)} = (1, 2, 4, 6, 7) \), we obtain \( \bar{t}_1 = 0.4; \bar{t}_2 = 0.9; \bar{t}_4 = 1.5; \bar{t}_5 = 2.3; \bar{t}_6 = 3.3; \bar{t}_7 = 4.8 \). Next we calculate the values of \( Z_1 = Z_2 = Z_4 = 0; Z_3 = 0.3; Z_5 = 0.3; Z_6 = 0.8 \).

3. From the set of \( A = \{1, 2, 4, 5\} \) let us choose the request for repair with the largest \( t_k \). In this case \( k = 5 \). So, we obtain sequence of \( \pi^{(5)} = (1, 2, 4, 6, 7) \). Then the values of \( \bar{t}_1 = 0.4; \bar{t}_2 = 0.9; \bar{t}_4 = 1.5; \bar{t}_6 = 2.5; \bar{t}_7 = 4 \) are calculated, for which all of \( Z_k = 0 \).

Hence, \( \pi^* = (1, 2, 4, 6, 7, 3, 5) \) is an optimal sequence of servicing the requests for repair.

Five requests for repair will be served for the established deadline.

3. Algorithm for design of the optimal sequence for the requests servicing

Let us consider the problem of design of the optimal sequence for servicing the requests for repair according to the criterion \( F_n = \sum_{k=1}^{n} q_k(x) \), where \( q_k(x) = \max (x - T_k, 0) \), \( k = 1, n \), \( x \) – is a moment for completion of servicing the request.

Let’s consider the description of the algorithm.

Step 1. The received requests for repair are numbered in the order of non-decrease of \( T_k \) values.

The a sequence \( \pi^j(j_1, j_2, \ldots, j_n) \) \( T_{j_i} \leq T_{j_i+1} = \bar{t}_n - T \) is obtained. If all of the requests in the process of requests service within the sequence of \( \pi^j \) are served in time meaning that a condition (1) is executed

\[
\sum_{k=1}^{n} t_{jk} \leq T_{j_k} \text{ or } t_{jk} + T_{j_k} \geq T = \sum_{k=1}^{n} t_k
\]
Then $\pi^T$ is an optimal sequence.

Step 2. The received requests for repair are numbered in the order of non-decrease of $t_k$ values. Then a sequence of $\pi^t = (i_1, i_2, \ldots, i_n)$, $t_i \leq t_{i+1}$, $j = 1, n - 1$ is obtained.

If during the service of the requests within the sequence of $\pi^t$ all of the requests are served in time, meaning that the condition (2) is executed

$$T_{ik} \leq T_{ij} \text{ or } T_{ik} \leq \sum_{l=1}^{k-1} t_{il} + t_{lj}, 1 \leq k < j \leq n$$

Then $\pi^t$ is the optimal sequence.

Step 3. If the sequences $\pi^t$ and $\pi^i$ do not satisfy sufficient conditions of the optimality then from the set $N$ a request for repair $l_n \text{ } t_{ln} = \max t_k, k \in N$ is chosen.

If condition (3) is chosen

$$T_k \leq \max(t_{ln}, T_{ln}), k \in N$$

then within the optimal sequence this request is served the last one. Request $l_n$ is excluded from set $N$, it is considered $N_l = N \setminus l_n$, and request $l_{n-1} \text{ } t_{ln-1} = \max Tt_k, k \in N_p$ is chosen.

If $T_k \leq \max (t_{ln-1}, T_{ln-1}), k \in N_l$, then the request $l_{n-1}$ within the optimal sequence is served the last but one.

Next, a set of $N_2 = N_l \setminus l_{n-1}$ is considered. As a result of the iteration process one can obtain a set of the requests $\{l_{n-p+1}, \ldots, l_n\}$, that are served last.

A set of the remained requests for the repair is denoted by $N_p$.

Step 4. Request $l_{n-p}$ is chosen with $T_{ln-p} = \max Tt_k, k \in N_p$ If the condition (4) is satisfied

$$t_{ln-p} + T_k \geq \sum_{l \in N_p} t_l, k \in N_B$$

then within the optimal sequence the requests from $N_p$ set are served first in the order of non-decrease values of $T_k$.

Step 5. If no one of the requests for repair does satisfy condition (4) then a set $M=N_p$ is considered.

Set $M$ is divided by two subsets $M_1$ and $M_2$ in the following way: if for $j \in M$ there exists such $i \in M$ that the interval $\Theta_1 = 0$, $\Theta_2 = \sum t_k$ is an interval of priority of $i \rightarrow j$ kind, then $j \in M_2$ while the rest requests belong to the set $M_1$.

All of the variants for servicing of requests in the set $M_1$ as the first ones within the optimal sequence of all the requests in $M$ set. To do so an arbitrary request $l_i \in M_1$ is chosen. The value of $T_k$ is reduced for all of $k \in M$, $k \neq i$, by the value of $t_l$. Then the requests are defined that can be served the last ones within the optimal sequence if condition (3) is satisfied and as the first ones if condition (4) is executed (block 15).

The value of the optimality criterion $F_2(\pi)$ is calculated.

Step 6. As a result, the finite number of sequences of requests for the repair is obtained and among them the desired sequence $\pi^*$ is chosen with the least value of the criterion $F_2(\pi)$.

Example 2. Let us construct the optimal (according to the criterion $F_2(\pi) = \sum_{k=1}^{n} \varphi_k(x)$) sequence $\pi^*$ of servicing the requests for repair with the account of $\varphi_k(x) = \max (x - T_k, 0), k = 1, n$. The values of $t_k$ and $T_k$ are presented in table 2.

| $k$ | $t_k$, hr. | $T_k$, hr. |
|-----|------------|------------|
| 1   | 1          | 1.2        |
| 2   | 1.5        | 4          |
| 3   | 1.2        | 3          |
| 4   | 0.4        | 0.5        |
| 5   | 0.6        | 1          |
| 6   | 0.5        | 1.3        |
| 7   | 0.8        | 2.5        |
1. Let us sort a set of \( N = \{k \} \) requests for repair by non-decrease of \( T_k \) values, and thus obtain \( \pi^T \) (4, 5, 1, 6, 7, 3, 2). Let us check the execution of the condition for optimal sequence of the requests for repair, i.e. conditions for servicing of requests in time, \( \sum_{k=1}^{n} t_k \leq T_n, \sum_{k=1}^{n} t_k = 6, T_2 = 4 \).

Sequence of \( \pi^T \) does not satisfy sufficient conditions of optimality.

2. Let us sort a set of requests for repair by non-decrease of the values \( t_k \), and thus obtain \( \pi^t \) (4, 6, 5, 7, 1, 3, 2).

Now we check execution of the condition for the optimal sequence of requests for repair \( T_i \leq T_j, 1 \leq i < j \leq k \). This condition is not satisfied. Hence, the sequence of \( \pi^t \) does not satisfy sufficient conditions of the optimality.

3. Now let us consider a set of \( N \) requests in order to choose the request for servicing of the last one in the optimal sequence.

We choose the request \( l_1 = 2 \). \( t_{l1} = \max T_k = 1.5, k \in N, N = l_5 \{k \} \ldots \). Since \( T_k \leq \max (t_2, T_2) = 4 \) for all of \( k \in N \), then the request 2 in the desired sequence of \( \pi^* \) can be served the last one.

From a set of \( N_1 = N \setminus \{2\} \) we choose the request \( l_3 = 3 \) with \( t_{l3} = \max t_k = 1,2, \ldots, k \in N_1 \).

Since \( T_k \leq \max (t_3, T_3) = 3 \) for all of \( k \in N_1 \), then the request 3 in \( \pi^* \) can be served the last but one. Thus, the requests for repair 3, 2 can be served the as the last ones in the desired sequence.

4. Let us consider a set of \( N_2 = N \setminus \{2,3\} \) in order to choose the request that will be served the first within the optimal sequence of requests in the order of non-decrease of \( T_k \). Among the requests for repair in \( N_2 \) set we choose the request \( l_1 = 7 \) with \( T_{l1} = \max T_k = 2.5, k \in N_2 \).

Since \( t_{l1} + T_7 \geq \sum_{i \in N_1} t_i \), 0.8 + 2.5 \( \geq \) 1 + 0.4 + 0.6 + 0.5 + 0.8, then within the optimal sequence of \( \pi^* \) the request for repair number 7 will be served as the fifth one.

Let us proceed to the problem of ordering in the service of the remaining requests for the repair.

Consider a set of \( M = N_2 \setminus \{7\} \) within the time interval \( \{0 \leq t \leq 2.5\} \).

1 variant. Let \( l_1 = 4 \). Let us decrease the value of \( T_k \) for all of \( k \in M, k \neq 4 \), by the value of \( t_k = 0.4 \) and then proceed to the consideration of the equivalent problem of the optimal servicing of sequences in the interval of \([0;2]\), the values of \( t_k \) and \( T_k \) are presented in table 3.

\[
\text{Table 3. Data on the requests for repair.}
\begin{array}{cccc}
  k & 5 & 1 & 6 \\
  \hline
  t_k & 0.6 & 1 & 0.5 \\
  T_k & 0.6 & 0.8 & 0.9 \\
\end{array}
\]

Let us choose the request \( l_4 = 1 \), \( t_{l4} = \max_{k \in \{1,5,6\}} t_k = 1 \). Since \( T_1 \leq \max (t_1, T_1) = 1 \) for \( k = 5, 6 \). Request 1 can be served the fourth in the order.

Choose the request \( l_2 = 6 \) with \( T_{l2} = \max_{k \in \{5,6\}} T_k = 0.9 \), since \( t_2 + T_6 \geq \sum t_i \), then the request 6 can be served the second one.

Thus, we obtain the sequence \((6, 5, 1)\) with the value of \( F_2(\pi) = 0 + 0.5 + 1.3 = 1.8 \).

2 variant. Let \( l_1 = 6 \). Decrease the value of \( T_k \) for all of \( k \in M, k \neq 6 \) by 0.5 and next proceed to the consideration of equivalent problem for the optimal servicing of requests within the interval of \((0;2)\), the values of \( t_k \) and \( T_k \) are presented in table 4.

\[
\text{Table 4. Data on the requests for repair.}
\begin{array}{cccc}
  k & 4 & 5 & 1 \\
  \hline
  t_k & 0.4 & 0.6 & 1 \\
  T_k & 0 & 0.5 & 0.7 \\
\end{array}
\]
Let us choose the request $l_4 = 1, t_{l_4} = \max_{k \in \{1,4,5\}} t_k = 1$. Since $T_k \leq \max (t_1, T_1) = 1$ for $k = 4, 5$. The request 1 can be served the fourth one in the order.

Choose the request $l_5 = 5, t_{l_5} = \max_{k \in \{4,5\}} t_k = 0$, since $T_k \leq \max (t_5, T_5) = 0.6$ for $k = 4$, the request 5 can be served the third one in the order.

Thus, we obtain sequence $(4, 6, 1)$ with the value of $F_2^\Sigma (\pi) = 0.4 + 0.5 + 1.3 = 2.2$

Let us choose the variant $1 -$ sequence $(6, 5, 1)$, $l_1 = 4$ with the least value of $F_2^\Sigma (\pi) = 1.8$

Then the desired sequence of servicing the requests for repair is the sequence of $\pi^- = (4,6,5,1,7,3,2)$ and $F_2^\Sigma (\pi^-) = 3.7$.

4. Conclusion
Algorithm for servicing of the maximal number of the requests for repair within the established deadlines was designed in the work. Example of the practical application of the algorithm is provided.
Algorithm for servicing of the optimal sequence of requests for the repair obtained with the account of the least total time of delays during completion of service within the established deadlines was designed as well. Example of the practical application of the algorithm is presented.

References
[1] McNaughton R 1959 Scheduling with deadlines and loss functions Manag. Sci. 6(1) 1-12
[2] Rothkopf M 1966 Scheduling with random service times Manag. Sci. 12(9) 707-13
[3] Kovalen M Y, Shafanskij Y M and Tanaev V S 1989 Approximation scheduling algorithms: a survey Optimization: A Journal of Mathematical Programming and Operations Research 20(6) 859-78
[4] Lawler E L and Moore J M 1969 A functional education and its application to resource allocation and sequencing problems Manag. Sci. 16(1) 77-84
[5] Tanaev V S, Sotskov Y N and Strusevich V A 1994 Scheduling Theory. Multi-Stage Systems (London: Kluwer Academic Publishers)
[6] Moore J M 1968 An n – job, machine sequencing algorithm for minimizing the number of late jobs Manag. Sci. 15(1) 102-9
[7] Strum L B 1970 A simple optimality proof of Moore’s sequencing algorithm Manag. Sci. 17(1) 116-8
[8] Sidney J B 1973 An extention of Moore’s due date algorithm Lect. Notes Econ. and Math. Syst. 86 393-8
[9] Emmons H 1969 One – machine sequencing to minimize certain functions of job Operat. Res. 4 701-15
[10] Srinivasan V 1971 A hybrid algorithm for the one machine sequencing problem to minimize total tardiness Nav. Res. Log. Quart. 18(3) 317-27