Модел. и анализ информ. систем. Т. **22**, № **3** (2015) 356–371

DOI: 10.18255/1818–1015–2015–3–356–371

UDC 519.987

Scheduling Problems of Stationary Objects with the Processor in One-Dimensional Zone

Dunichkina N. A.*, Kogan D. I.**, Fedosenko Yu. S.*[1]

**Volga State University of Water Transport,*
Nesterova str., 5a, Nizhny Novgorod, 603005, Russia
***Moscow State University of Instrument Engineering and Informatics,*
Stromynka str., 20, Moscow, 107996, Russia

e-mail: nadezhda.dunichkina@gmail.com, kdi_41@mail.ru, fds@vgavt-nn.ru

received March 25, 2015

Keywords: scheduling, dynamic programming, Pareto concept, NP-complexity, multicriteria optimization

We consider the mathematical model in which an operating processor serves the set of the stationary objects positioned in a one-dimensional working zone. The processor performs two voyages between the uttermost points of the zone: the forward or direct one, where certain objects are served, and the return one, where remaining objects are served. Servicing of the object cannot start earlier than its ready date. The individual penalty function is assigned to every object, the function depending on the servicing completion time. Minimized criteria of schedule quality are assumed to be total service duration and total penalty. We formulate and study optimization problems with one and two criteria. Proposed algorithms are based on dynamic programming and Pareto principle, the implementations of these algorithms are demonstrated on numerical examples. We show that the algorithm for the problem of processing time minimization is polynomial, and that the problem of total penalty minimization is NP-hard. Correspondingly, the bi-criteria problem with the mentioned evaluation criteria is fundamentally intractable, computational complexity of the schedule structure algorithm is exponential. The model describes the fuel supply processes to the diesel-electrical dredgers which extract non-metallic building materials (sand, gravel) in large-scale areas of inland waterways. Similar models and optimization problems are important, for example, in applications like the control of satellite group refueling and regular civil aircraft refueling.

The article is published in the author's wording.

[1]The article prepared with a financial support of Russian Fund of Fundamental Research – project № 15-07-03141.

Introduction

The problems under study were posed when it was necessary to create computer-based systems for operating control of fuel supply to the floating diesel-electrical complexes or dredgers extracting non-metallic building materials (gravel, sand) in larger transport areas of inland waterways. One of the transport area operator's responsibilities is to work out the time schedule [1–6] reducing cost losses due to idling of both dredgers and a fuel supply tanker. In this paper we formulate optimization problems for the model in which the moving processor is to serve the set of stationary objects positioned within uniform one-dimensional working zone. The processor is assumed to do two-way voyages – the forward or direct one, during which few objects are served, and the return one, when the remaining objects are served. Individual penalty function is assigned to each object; it is a monotone increasing function associated with the time when servicing of the particular object is accomplished. The minimized criteria are the service completion time of all the objects involved and the total penalty. Similar models and optimization problems are important, for example, in applications like the control of satellite group refueling [7] and regular civil aircraft refueling [8].

1. Mathematical Model And Problems Formulation

There is an assumed set $O_n = \{o_1, o_2, \dots, o_n\}$ of the stationary objects within the working zone L of the operating processor P (fig. 1). The working zone is one-dimensional and finite; its initial point A is a start up point for the processor. Objects are supposed to be numbered in the order of their distances increasing from the point A; the end point B of the zone L is the location of the object o_n. Starting from the moment $t = 0$ the processor moves from the start up point A towards the end point B (forward voyage, let us denote it by λ_+), and having reached the end point, it moves back to the point A (return voyage, let us denote it by λ_-).

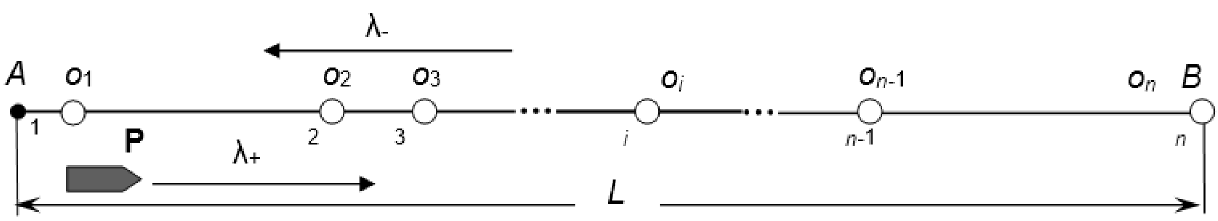

Fig. 1: Modelling single processor servicing the related objects.

During the cycle $\lambda_+\lambda_-$ the processor P performs single continuing service of group O_n-related objects: a few of them are served in voyage λ_+ , remaining objects – in voyage λ_-. Simultaneous servicing of two and more objects is prohibited.

With every object o_j we associate monotone non-decreasing penalty function of its service completion time; it represents the losses related to the service.

By $1, 2, \dots, n$ we denote segment L points where the objects $o_1, o_2, \dots, o_n$ are positioned correspondingly (the points n and B coincide); τ_j - object o_j service duration, r_j - ready date of the object o_j; $\varphi_j(t)$ - the object o_j penalty function (if servicing of the object o_j

is accomplished at the moment t then $\varphi_j(t)$ is a penalty for this particular object; $\gamma_{j-1,j}$ and $\gamma_{j,j-1}$ - the processor movement durations between $j-1$ and j in the voyages λ_+ and λ_- respectively; $j = \overline{1,n}$, here $\gamma_{0,1}$ and $\gamma_{1,0}$ - the processor movement durations between point A and the point 1 in the voyages λ_+ and λ_- correspondingly. The values τ_j, $\gamma_{j-1,j}$ and $\gamma_{j,j-1}$ are positive integers, r_j are non-negative integers.

Servicing strategy is an arbitrary subset of ascending indices $V = (i_1, i_2, \ldots, i_k)$ of the set $N = 1, 2, \ldots, n$. During the strategy realization the objects o_{i_m}, where $i_m \in V$, are served in voyage λ_+; the remaining objects of the set O_n are served in voyage λ_-. By $V^- = (i_{k+1}, i_{k+2}, \ldots, i_n)$ as defined by strategy V we denote the sequence of the objects served in voyage λ_-, the indices in V^- are listed in the diminishing order. The sequences V and V^- do not contain equal elements. To be explicit, we assume that the object o_n is served upon completion of voyage λ_+, hence $n \in V$. Let us note that $V^- = \varnothing$ if and only if $V = (1, 2, \ldots, n)$. It is evident that the number of different servicing strategies is equal to 2^{n-1}. We assume that service time schedules to apply the strategy V are the tuples as follows:

$$\rho = \langle (i_1, a_{i_1}, b_{i_1}), (i_2, a_{i_2}, b_{i_2}), \ldots, (i_k, a_{i_k}, b_{i_k}), \ldots, (i_n, a_{i_n}, b_{i_n}) \rangle\,,$$

where $V = (i_1, i_2, \ldots, i_k)$, $V^- = (i_{k+1}, i_{k+2}, \ldots, i_n)$, $i_k = n$, a_{i_m} and b_{i_m} - servicing start up and completion time for the object o_{i_m} respectively, $m = \overline{1,n}$; $a_{i_1} \geq \gamma_{0,i_1}$, $b_{i_1} = a_{i_1} + \tau_{i_1}$; $a_{i_2} \geq b_{i_1} + \gamma_{i_1,i_2}$, $b_{i_2} = a_{i_2} + \tau_{i_2}$; $\ldots$; $a_{i_n} \geq b_{i_{n-1}} + \gamma_{i_{n-1},i_n}$, $b_{i_n} = a_{i_n} + \tau_{i_n}$. Further we denote the object o_x servicing start up and completion time by $S_x(\rho)$ and $C_x(\rho)$ correspondingly, the values being depended on the schedule ρ.

By $K_(\rho)$ we denote the total penalty for all objects under service during the schedule ρ implementation, $K(\rho) = \left\{ \sum\limits_{j=1}^{n} \varphi_j(C_j(\rho)) \right\}$. By $T(\rho)$ we denote the time when the processor returns to the initial point after service accomplishment involving the objects according to schedule ρ. For arbitrary schedule ρ we have:

$$T(\rho) = b_{i_n} + \gamma_{i_n,0}. \tag{1}$$

The schedule ρ is called r-feasible if during its implementation all ready dates for the objects are observed. The set of all r-feasible schedules (each of them implements some servicing strategy) is assumed here as R; the set of all r-feasible schedules which implements the strategy V will be denoted as $R(V)$. It is obvious that any set $R(V)$ is nonempty.

Further, we consider the following two problems.

Problem 1. $\min\limits_{\rho \in R} T(\rho)$.

Problem 2. $\min\limits_{\rho \in R} \{K(\rho), T(\rho)\}$.

The problem 1 is to construct the schedule optimal by processing time. In bi-criteria problem 2 the first criterion is total penalty for the objects, the second one - servicing cycle duration. For problem 2 we will use the Pareto concept, which implies the synthesis of the total set of the efficient estimates, simultaneously providing the opportunity to determine the problem solution that assures any efficient estimate [9–12].

For problems 1 and 2 we will further construct the respective algorithms of polynomial and exponential computational complexity. Both algorithms are based on dynamic programming [13,14]. We will further show, that problem 2 is fundamentally intractable. This intractability follows from the NP-hardness [15–17] of a one-criterion problem below.

Problem 3. $\min_{\rho \in R} K(\rho)$.

We will show below that problem 3 is NP-hard even in a particular case, when all functions $\varphi_j(t)$, $j = \overline{1,n}$, are linear. If we construct the set of efficient estimates for problem 2 this will inevitably lead us to the solution of problem 3.

2. Compact schedules and 0-schedules

Schedule ρ, $\rho \in R$, is called compact, if in-between stops of the processor during the cycle $\lambda_+\lambda_-$ are only related to the objects service in their locations, and to their expectancy for the ready dates to come. When constructing the compact schedule for arbitrary strategy $V = (i_1, i_2, \ldots, i_k)$, the servicing start and completion times a_{i_m} and b_{i_m} of the object o_{i_m}, $m = \overline{1,n}$, are calculated consecutively to the extent that the parameter m grows, the following formulas being:

$$a_{i_1} = \max(\gamma_{0,i_1}, r_{i_1}); \tag{2}$$

$$b_{i_m} = a_{i_m} + \tau_{i_m}, m = \overline{1,n}; \tag{3}$$

$$a_{i_{\chi+1}} = \max(b_{i_\chi} + \gamma_{i_\chi, i_{\chi+1}}, r_{i_{\chi+1}}), \chi = \overline{1, n-1} \tag{4}$$

We will denote the compact schedule implementing arbitrary strategy V by $\rho_k(V)$. Specified schedule is defined unambiguously.

It should be noted that the processor which started the forward voyage relatively late (for example, at the moment $\max_p p$) can serve the objects of set $O_n = \{o_1, o_2, \ldots, o_n\}$ without intermediate idlings which arise from the need to observe the ready dates. By $t_0(V)$ we denote the minimal forward voyage start time so that the processor can further serve all the objects of the set O_n according to strategy V without intermediate idlings as a result of ready dates $r_1, r_2, \ldots, r_n$. We will call this servicing mode as "0-mode and related schedule is named here as "0-schedule". The 0-schedule implementing the strategy V will be denoted as $\rho_0(V)$.

It is evident that $t_0(V)$ is a total idle time of the processor in waiting for the ready dates $r_1, r_2, \ldots, r_n$ during the schedule $\rho_k(V)$ implementation. For fixed strategy V the value $t_0(V)$ is defined as follows.

1. We sequentially calculate the values a_{i_m} and b_{i_m} for the schedule $\rho_k(V)$ using the formulas(2) – (4) with a gradual parameter m growth, $m = \overline{1,n}$;
2. Time losses T^* are then calculated for direct servicing of all the objects involved, as well as for the processor movement from the point 0 to the point n and then

back to the last object to be serviced in the strategy:

$$T^* = \begin{cases} \sum_{j=1}^{n} \tau_j + \gamma_{0,n} + \gamma_{n,i_n}, & \text{if the set of the objects served in the backward voyage} \\ & \text{is nonempty;} \\ \sum_{j=1}^{n} \tau_j + \gamma_{0,n}, & \text{in opposite case} \end{cases}$$

3. We set $t_0(V) = b_{i_n} - T^*$.

For 0-schedule $\rho = \langle (i_1, a'_{i_1}, b'_{i_1}), (i_2, a'_{i_2}, b'_{i_2}), \ldots, (i_k, a'_{i_k}, b'_{i_k}), \ldots, (i_n, a'_{i_n}, b'_{i_n}) \rangle$, defined by the servicing strategy $V = (i_1, i_2, \ldots, i_k)$, we have the following relations:

$$a'_{i_1} = t_0(V) + \gamma_{0,1}; \tag{5}$$

$$b_{i_m} = a'_{i_m} + \tau_{i_m}, m = \overline{1, n}; \tag{6}$$

$$a'_{i_{\chi+1}} = b'_{i_\chi} + \gamma_{i_\chi, i_{\chi+1}}, \chi = \overline{1, n-1} \tag{7}$$

For the given initial data, 0-schedule implementing the arbitrary strategy V can be uniquely defined. We should note, that if $t_0(V) = 0$, the schedules $\rho_0(V)$ and $\rho_k(V)$ are identical.

Theorem 1. *The schedule $\rho_k(V)$ minimizes value of $K(\rho)$ on the set $R(V)$; both schedules $\rho_0(V)$ and $\rho_k(V)$ minimize the values of the $T(\rho)$ on the set $R(V)$.*

The theorem statements are easily proved by contradiction.

3. Problem 1 solving algorithm

According to theorem 1, problem 1 permits the following equivalent form.

Problem 4. $\min_V T(\rho_0(V))$.

From the definition of 0-schedule it follows that:

$$T(\rho_0(V)) = t_0(V) + \sum_{j=1}^{n} \tau_j + \sum_{j=1}^{n-1} \gamma_{j,j+1} + \sum_{j=n}^{1} \gamma_{j,j-1}. \tag{8}$$

Thus, the problem of the criterion $T(\rho_0(V))$ minimization and, equally, the criterion $T(\rho_k(V))$ minimization, reduces to the minimization of the value $t_0(V)$:

$$\min_V t_0(V). \tag{9}$$

Having defined the subset V as optimal for problem (9), we will easily then construct the optimal servicing schedule for problem 1.

Let us denote as $D(k)$ the start up marginal momentum, when the processor in point k can serve all the objects $\{o_k, o_{k+1}, \ldots, o_n\}$ in the 0-mode during the subsequent implementation of the forward or direct voyage (from point k) and then the return voyage, i.e. without idlings time due to the ready dates $r_k, r_{k+1}, \ldots, r_n$, here $k \in \{1, 2, \ldots, n\}$. Together with the values $D(k)$ calculation, we will consecutively construct the strategy V_D which assures these values.

It is evident that

$$D(n) = r_n. \tag{10}$$

Sequence V_D being composed is initially assumed as a single element n. We select the following notation:

$$W^*(k) = (\tau_k + \tau_{k+1} + \ldots + \tau_n) + (\gamma_{k,k+1} + \gamma_{k+1,k+2} + \ldots + \gamma_{n-1,n} + \gamma_{n,n-1} + \ldots + \gamma_{k+1,k});$$

thus, $W^*(k)$ is the total time of the direct servicing of the objects $\{o_k, o_{k+1}, \ldots, o_n\}$ and the processor movements from the point k to the point n and from the point n to the point k in the direct and return voyages respectively.

There is an alternative for each object from the set $\{o_1, o_2, \ldots, o_{n-1}\}$: it can be served either in the direct or in the return voyage. Assuming that there are no idlings, servicing of the object o_{n-1} can start in the direct voyage at the moment t'_{n-1} if and only if $(t'_{n-1} \geq r_{n-1})\&(t'_{n-1} + \tau_{n-1} + \gamma_{n-1.n} \geq D(n))$. The minimal possible value t'_{n-1} which meets the above constraints is equal to $\max\{r_{n-1}, D(n) - (\tau_{n-1} + \gamma_{n-1,n})\}$.

Let us assume that servicing of the object o_{n-1} is performed in the return voyage. With no idlings assumed, the processor skips the servicing of the object o_{n-1} in the direct voyage, and can start moving from the point $n-1$ towards the point n at the moment t''_{n-1} if and only if $(t''_{n-1} + \gamma_{n-1,n} \geq D(n))\&(t''_{n-1} + W^*(n-1) - \tau_{n-1} \geq r_{n-1})$. The minimal possible value t''_{n-1} in this case is equal to $\max\{D(n) - \gamma_{n-1,n}, r_{n-1} - (W^*(n-1) - \tau_{n-1}), 0\}$ resulting in:

$$D(n-1) = \min \begin{bmatrix} \max\{r_{n-1}, D(n) - (\tau_{n-1} + \gamma_{n-1,n})\}, \\ \max\{D(n) - \gamma_{n-1,n}, r_{n-1} - (W^*(n-1) - \tau_{n-1}), 0\} \end{bmatrix}. \tag{11}$$

Index $n-1$ is included in the sequence V_D if $D(n-1) = \max\{r_{n-1}, D(n) - (\tau_{n-1} + \gamma_{n-1,n})\}$.

Let us assume that for arbitrary $k \in \{2, 3, \ldots, n-1\}$ the value $D(k)$ has been obtained. With no idlings assumed servicing of the object o_{k-1} during the direct voyage can start at the moment t'_{k-1} if and only if $\left(t'_{k-1} \geq r_{k-1}\right)$ & $\left(t'_{k-1} + \tau_{k-1} + \gamma_{k-1,k} \geq D(k)\right)$.

The minimal possible value t'_{k-1}, for which the given constrains are met, is equal to $\max\{r_{k-1}, D(k) - (\tau_{k-1} + \gamma_{k-1,k})\}$. Let servicing of the object o_{k-1} be performed in the return voyage. With no idlings assumed the processor skips the servicing of the object o_{k-1} in the direct voyage and can start moving from the point $k-1$ towards the point k at the moment t''_{k-1} if and only if $(t''_{k-1} + \gamma_{k-1,k} \geq D(k))\&(t''_{k-1} + W^*(k-1) - \tau_{k-1} \geq r_{k-1})$.

Thus the minimal possible value t''_{k-1} is equal to $\max\{D(k) - \gamma_{k-1,k}, r_{k-1} - (W^*(k-1) - \tau_{k-1}), 0\}$ resulting in:

$$D(k-1) = \min \begin{bmatrix} \max\{r_{k-1}, D(k) - (\tau_{k-1} + \gamma_{k-1,k})\}, \\ \max\{D(k) - \gamma_{k-1,k}, r_{k-1} - (W^*(k-1) - \tau_{k-1}), 0\} \end{bmatrix}, \tag{12}$$

$k \in \{2, 3, \ldots, n-1\}$.

We include index $k-1$ in sequence V_D if $D(k-1) = \max\{r_{k-1}, D(k) - (\tau_{k-1} + \gamma_{k-1,k})\}$ and accomplish strategy V_D construction when calculations using formula (12) have been made with parameter k values consequently decreasing.

By $D(0)$ we denote the minimal start time of the movement from point 0, when the processor can serve all the objects of the set $O_n = \{o_1, o_2, \dots, o_n\}$ during the cycle $\lambda_+\lambda_-$ in the 0-mode, i.e. without idlings due to the ready dates. It means that $D(0) + \gamma_{0,1} \geq D(1)$. Hence we have:

$$D(0) = \max\ (D(1) - \gamma_{0,1}, 0). \qquad (13)$$

The equations (10) – (13) are dynamic programming relations which allow to consecutively define values $D(n), D(n-1), D(n-2), \dots, D(0)$. According to the introduced definitions of the value $D(0)$ and the function $t_0(V)$, we obtain the equation:

$$D(0) = \min_V t_0(V) = t_0(V_D) \qquad (14)$$

Strategy V_D constructed is the optimal solution for problem 4. Corresponding schedules $\rho_0(V_D)$ and $\rho_k(V_D)$ are optimal for problem 1.

It should be noted that calculation of every succeeding value $D(k)$ (as the argument decreases) involves few operations. Hence, the proposed algorithm to solve problem 1 is functioning in linear time (from n).

E x a m p l e 1. *Optimal schedule related to criterion $K_1(\rho)$ is to be obtained with objects o_1, o_2, o_3 and o_4 located at points* 1, 2, 3 *and* 4 *respectively to be served;* $\gamma_{0,1} = \gamma_{1,0} = 2$, $\gamma_{1,2} = \gamma_{2,1} = 1$, $\gamma_{2,3} = \gamma_{3,2} = 10$, $\gamma_{3,4} = \gamma_{4,3} = 1$, $r_1 = 1$, $r_2 = 10$, $r_3 = 12$, $r_4 = 15$, $\tau_1 = \tau_2 = \tau_3 = \tau_4 = 1$.

Firstly, we calculate values $W^*(k)$: $W^*(1) = 28$, $W^*(2) = 25$, $W^*(3) = 4$, $W^*(4) = 1$. According to formula (10), we set: $D(4) = 15$. The sequence V_D is initially assumed to be of a single element n equalling here to 4. Then according to formula (11) we obtain: $D(3) = \min\ [\max\{12, 15-(1+1)\}, \max\{15-1, 12-(4-1), 0\}] = 13$; index 3 being included in sequence V_D. According to formula (12) when $k = 3$ we receive: $D(2) = \min\ [\max\{10, 13-(1+10)\}, \max\{13-10, 10-(25-1), 0\}] = 3$; with index 2 being not included in sequence V_D. Then according to the same formula with $k = 2$ we get: $D(1) = \min[\max\{1, 3-(1+1)\}, \max\{3-1, 1-27, 0\}] = 1$; index 1 being included in sequence V_D. Finally, according to formula (13) we define $D(0) = 0$. In this case the optimal 0-schedule implementation starts from the moment 0; at the same time this schedule is compact. It is easy to define, that $\rho_0(V_D) = \rho_k(V_D) = \langle(1, 2, 3), (3, 14, 15), (4, 16, 17), (2, 28, 29)\rangle$. According to (1), the optimal criterion value for problem 1 is equal to 32.

4. Problem 2 solving algorithm

According to theorem 1, we can replace the problem under study $\min_{\rho \in R}\{K(\rho), T(\rho)\}$ by the following equivalent problem.

Problem 5. $\min_V\{K(\rho_k(V)), T(\rho_k(V))\}$.

Let us denote problem 5 by symbol Z; and the required set of the efficient estimates pertaining to the problem will be assumed as E. We will use a multi-criteria dynamic programming method [18–20] to synthesize this set.

Let us consider the set of particular problems $Z(k, t)$; problem $Z(k, t)$ is thought as a situation when the processor during voyage λ_+ arrives at point k at the moment t; the minimized criteria being:

– total penalty for the objects from the set $\{o_k, o_{k+1}, \dots, o_n\}$;

– time when processor leaves the point k in voyage λ_-.

Thus, the estimate (a, b) obtained for $Z(k, t)$ means, that the total penalty for objects $\{o_k, o_{k+1}, \dots, o_n\}$ is equal to a, and the processor leaves the point k during voyage λ_- at the moment b.

By $eff(M)$ we denote a set of efficient in set M estimates; the estimate (a, b) from M is efficient if there is no such estimate (a', b') in M so that $a' \le a$ and $b' \le b$ and at least one of the given inequalities is strict inequality. By $E(k, t)$ we denote a set of efficient estimates pertaining to problem $Z(k, t)$, where $k = \overline{1, n}$.

Evidently,

$$E(n, t) = (\varphi_n(\max(t, r_n) + \tau_n), \max(t, r_n) + \tau_n). \tag{15}$$

Let us assume that the sets $E(k+1, t)$ have already been constructed for all possible values of parameter t. We need to construct the sets $E(k, t)$.

Let a priori be known that the objects of the set $\{o_{k+1}, o_{k+2}, \dots, o_n\}$ are served with estimate (p, q) and the processor arrives at the point k at the moment t during voyage λ_+ and further it serves the object o_k. In this case the estimate of servicing the objects $\{o_k, o_{k+1}, \dots, o_n\}$ is

$$A(t, k, p, q) = (\varphi_k(\max(t, r_k) + \tau_k) + p, q + \gamma_{k+1,k}). \tag{16}$$

Since servicing of object o_k accomplishes at the moment $\mu_k = \max(t, r_k) + \tau_k$, the processor arrives at the point $k+1$ at the moment $\mu_k^* = \mu_k + \gamma_{k,k+1}$. Further servicing of the set $\{o_{k+1}, o_{k+2}, \dots, o_n\}$ can be effected with the estimates from the set $E(k+1, \mu_k^*)$; implementation of the estimates that do not belong to this particular set is obviously impractical. For the set $\{o_{k+1}, o_{k+2}, \dots, o_n\}$ we obtain the set of the estimates

$$P(k, t) = \{A(t, k, p, q) : (p, q) \in E(k+1, \mu_k^*)\} \tag{17}$$

provided that servicing of the object o_k is performed in voyage λ_+.

If on arriving at point k processor postpones object o_k servicing till voyage λ_-, and servicing of objects $\{o_{k+1}, o_{k+2}, \dots, o_n\}$ is performed with the estimates (p', q') then the estimate of object servicing from set $\{o_k, o_{k+1}, \dots, o_n\}$ is as follows:

$$B(t, k, p', q') = (p' + \varphi_k(\max(q' + \gamma_{k+1,k}, r_k) + \tau_k), \max(q' + \gamma_{k+1,k}, r_k) + \tau_k). \tag{18}$$

In the considered case the processor performing voyage λ_+ arrives at point $k+1$ at the moment $\nu_k^* = t + \gamma_{k,k+1}$. Further servicing of the set $\{o_{k+1}, o_{k+2}, \dots, o_n\}$ can be performed with the estimates from the set $E(k+1, \nu_k^*)$; implementation of the estimates that do not belong to this set is obviously impractical. For the set $\{o_k, o_{k+1}, \dots, o_n\}$ we obtain the following set of the estimates on the assumption that object o_k servicing is effected in voyage λ_-:

$$Q(k, t) = \{B(t, k, p', q') : (p, q) \in E(k+1, \nu_k^*)\}. \tag{19}$$

It is evident that

$$E(k,t) = eff(P(k,t) \cup Q(k,t)),\ \ k = n-1, n-2, \dots, 1. \tag{20}$$

The computational process using formulas (15) – (20) implies a consequent search of sets $E(k,t)$ when index k is decreasing. This finally leads to the construction of set $E(1, \gamma_{0,1})$.

To obtain target set E of the efficient estimates pertaining to problem 2, we need to add vector $(0, \gamma_{1,0})$ to each vector of set $E(1, \gamma_{0,1})$:

$$E = \{x = (x_1, x_2) : x_1 = y_1, x_2 = y_2 + \gamma_{1,0},\ \text{where } y = (y_1, y_2) \in E(1, \gamma_{0,1})\}. \tag{21}$$

Prior to calculations based on relations (15) – (21), for each k, $k = \overline{1,n}$ we need to define sets Θ_k of possible processor arrival moments t to point k during voyage λ_+. Only for the values t belonging to Θ_k, we need to construct sets $E(k,t)$ when calculating with the help of recurrent relations (15) – (20). The sets Θ_k are defined to the extent that index k values diminish. It is obvious, that when $k = 1$ the only possible value of t is $\gamma_{0,1}$, i.e. $\Theta_1 = \{\gamma_{0,1}\}$. Let us denote as $M_{k+1}(N_{k+1})$ the set of the possible values of the processor arrival moments to point $k+1$ in voyage λ_+ when object o_k was respectively served during $\lambda_+(\lambda_-)$. It is obvious that

$$M_{k+1} = \{\mu_k^* : \mu_k^* = \max(\theta, r_k) + \tau_k + \gamma_{k,k+1}, \theta \in \Theta_k\};$$

$$N_{k+1} = \{\nu_k^* : \nu_k^* = \theta + \gamma_{k,k+1}, \theta \in \Theta_k\};$$

$$\Theta_{k+1} = \{M_{k+1} \cup N_{k+1}\}, \text{where } k = 1, 2, \dots, n-1. \tag{22}$$

E x a m p l e 2. *It is required to discover a full set of efficient estimates pertaining to problem 2 with the parameters values and the penalty functions given in Table 1.*

Table 1: Modelling parameters, Example 2

j	τ_j	r_j	$\gamma_{j-1,j}$	$\gamma_{j,j-1}$	$\varphi_j(t)$
1	1	1	2	2	0
2	1	10	1	1	$\max\{t-3, 0\}$
3	1	14	10	10	$\max\{10(t-15), 0\}$
4	1	14	1	1	$\max\{15(t-16), 0\}$

Firstly, for each value $k(k = \overline{1,n})$ we need to find the sets Θ_k of possible moments t of the processor arrival to point k in voyage λ_+. It is apparent that the only possible value of t when $k = 1$ is $\gamma_{0,1} = 1$, i.e. $\Theta_1 = \{2\}$. According to (22), we find that: $\Theta_2 = \{3, 4\}, \Theta_3 = \{13, 14, 21\}, \Theta_4 = \{14, 15, 16, 22, 23\}$.

Then with the help of formulas (15) –(20) we calculate values $E(k,t)$. The established estimates are shown in Table 2. For each estimate the triple index is given: the number of the estimate (revealed when calculating), the estimate number used to have the current estimate constructed, and the voyage in which the object o_k is served. For example, the record $(a,b)_{i,j,+}$ means that the estimate (a,b) of number i is constructed from the estimate having number j, and object o_k is served in voyage λ_+. Similarly the record $(a,b)_{i,j,-}$ means that the estimate (a,b) of number i is constructed from the estimate having number j, and the object o_k is served in voyage λ_-. This information will be further used when we construct servicing strategy, which assures particular criteria values.

Table 2 filled in from right to left. Firstly, we fill in the cells of the fourth column ($k = 4$) with the help of (15). Here we assume that all estimates for $k = 4$ are constructed from the dummy estimate of number 0 and we take into account that the last object is always served in voyage λ_+. We thus gain as follows:

$$E(4,14) = \{(\varphi_4(\max(14,r_4) + \tau_4), \max(14,r_3) + \tau_3)\} = \{(\varphi_4(14), 15)\} = \{(0,15)\}_{1,0,+}.$$

Similarly, we define values:

$E(4,15) = \{(0,\ 16)\}_{2,0,+}$, $E(4,16) = \{(15,\ 17)\}_{3,0,+}$, $E(4,22) = \{(105,\ 23)\}_{4,0,+}$, $E(4,23) = \{(120,24)\}_{5,0,+}$.

Filling in the required cells for k equaling to 3, 2 and 1 is executed with the help of (16) – (20). Example of set $E(3,13)$ construction illustrates the calculations based on the above relations. $E(3,13) = eff(P(3,13) \cup Q(3,13))$ in accord with (20). Set P(3,13) is defined starting with value μ_3^* calculation:

$$\mu_3^* = \mu_3 + \gamma_{3,4} = \max(t,r_3) + \tau_3 + \gamma_{3,4} = \max(13,14) + 1 + 1 = 16.$$

Thus calculating value $P(3,13)$ we will use the only estimate $(15,17)$ of set $E(4,16)$. According to (16) and (17) we obtain $P(3,13) = \{(\varphi_3(\max(13,r_3)+\tau_3)+15, 17+\gamma_{4,3})\} = \{(\varphi_3(15)+15, 18)\} = \{(15,18)\}$.

Set $Q(3,13)$ is defined starting with value ν_3^* calculation: $\nu_3^* = t + \gamma_{3,4} = 13+1 = 14$. Thus, during calculations we will use estimates from set $E(4,14) = \{(0,15)\}$. According to (18) and (19) we get:

$Q(3,13) = \{(\varphi_3(\max(15+\gamma_{4,3}, r_3)+\tau_3)+0, \max(15+\gamma_{4,3}, r_3)+\tau_3)\} = \{(\varphi_3(17), 17)\} = \{(20,17)\}$.

Finally we obtain: $E(3,13) = eff\big((15,18) \cup (20,17)\big) = \{(15,18),(20,17)\}$.

The consecutive number 6 is assigned to the estimate $(15,18)$, which is constructed on the basis of the estimate having number 3, the calculations with the help of (17) correspond to object o_3 servicing in voyage λ_+. The record has been entered in the table cell as $(15,18)_{6,3,+}$.

Succeeding number 7 is assigned to the estimate $(20,17)$, the estimate being constructed from the estimate of number 1, and the calculation using formula (19) corresponds to object o_3 servicing in voyage λ_-. The record has been entered in the table cell as $(20,17)_{7,1,-}$.

The remaining estimates entered in table 2 have been calculated in a similar manner. The empty cells of Table 2 correspond to the unfeasible pairs (k,t).

Thus we have revealed that set $E(1,2)$ contains one estimate only. According to (21), the single efficient estimate in the considered example is $(41,32)$; thus, there has been defined an optimal strategy for both criteria considered here. The estimate indices that have been entered in Table 2 allow easy finding the strategy, which generates the estimate: objects 1, 3, 4 to be served in the direct voyage; object 2 to be served in the return voyage.

Table 2: $E(k,t)$ values

$t\backslash k$	1	2	3	4
2	$(41,30)_{13,12,+}$			
3		$(41,29)_{10,6,-}$ $(45,28)_{11,7,-}$		
4		$(41,29)_{12,6,-}$		
13			$(15,18)_{6,3,+}$ $(20,17)_{7,1,-}$	
14			$(15,18)_{8,3,+}$	$(0,15)_{1,0,+}$
15				$(0,16)_{2,0,+}$
16				$(15,17)_{3,0,+}$
21			$(190,25)_{9,5,+}$	
22				$(105,23)_{4,0,+}$
23				$(120,24)_{5,0,+}$

To estimate the computational complexity of the proposed algorithm we denote the maximum of the values $\gamma_{j-1,j}$, $\gamma_{j,j-1}$, τ_j, $j=\overline{1,n}$, by Q, and the maximum of the values r_j by Q^*, $j=\overline{1,n}$. It is evident that servicing objects of the set $O_n=\{o_1,o_2,\dots,o_n\}$ can be accomplished not later than $M=3Q+Q^*$. The second coordinate of every estimate taken from any set $E(k,t)$ does not exceed M. Hence there are not more than M estimates in each set $E(k,t)$. The number of estimates of each set $P(k,t)$ and $Q(k,t)$ constructed for $E(k,t)$ calculation also does not exceed M. Consequently, we need no more than linearly depending on M number of the elementary operations for each set $E(k,t)$ synthesis. The first argument in the sets $E(k,t)$ can take n values, the second argument – no more than M values. The upper bound for the number of constructed sets $E(k,t)$ is the product nM. Thus, the number of elementary operations necessary to construct the set E is bounded above by the value of nM^2 order. So the proposed algorithm of efficient estimates synthesis in problem 2 is pseudo-polynomial.

5. Computational complexity of the problem 3

It is known [21], that if all ready dates r_j are equal to zero and the individual penalty functions $\varphi_j(t)$ are linear, $j=\overline{1,n}$, then problem 3 has polynomial solution. However, one cannot avoid the fact as follows.

Theorem 2. *If all individual penalty functions are linear and there exists the single object with non-zero ready date, then problem 3 is NP-hard.*

The proof is that NP-complete partition problem [15] has to be polynomially-time reduced to problem 3 that satisfies the theorem conditions. The partition problem is given as follows. There exists a finite set of natural numbers $W=\{w_1,w_2,\dots,w_n\}$; the question arises whether it is possible to split this set into two disjoint subsets so that the sum of the numbers from the first subset is equal to the sum of the numbers from the second subset. The positive answer obviously necessitates the condition under which the number from the set W does not exceed the half of the sum of all numbers . It should be also noted that the problem remains NP-complete in the case when all elements of the set W are even numbers. Further we assume that $\sum_{i=1}^{n} w_i = 2U$, all w_i are even numbers and $w_i < U$, $i = \overline{1, n}$.

The initial data of the partition problem allow us to construct a problem where processor P during the successive voyages λ_+ and λ_- is to serve stationary objects $o_1, o_2, \dots, o_{n+2}$ located in one-dimensional working zone L; the object o_i is assumed to be located at point i, the values $\gamma_{i-1,i}, \gamma_{i,i-1}, i = \overline{1, n+2}$ are purported equaling to 1. The objects o_1 and o_{n+2} of the set $O_{n+2} = \{o_1, o_2, \dots, o_{n+1}, o_{n+2}\}$ are considered as "significant"; the remaining ones being assumed "ordinary". Servicing duration of an ordinary object o_{i+1} is assumed to be equal to w_i, $i = \overline{1, n}$; hence the total time expenditure for servicing all ordinary objects is equal to $2U$. Servicing periods of significant objects are the following: $\tau_1 = U+1, \tau_{n+2} = 1$. Individual penalty functions for significant objects are defined by the formulas: $\varphi_1(t) = t$; $\varphi_{n+2}(t) = Dt$, where D is sufficiently large constant, its value to be defined below. The individual penalty functions of the ordinary objects are supposed to be identically equal to zero. Each object o_j, $j = \overline{1, n+1}$ is assumed to be ready for servicing starting from the moment $t = 0$; assuming that $r_{n+2} = (n+2) + U$.

Servicing of the significant object o_{n+2} may commence at the moment r_{n+2} if and only if the processor consumes the time that does not exceed U envisaged for servicing the remaining objects in the direct voyage, bearing in mind that all movements of the processor in this voyage use up $n+2$ units of time.

Let us assume that servicing of the object o_{n+2} commences at the moment r_{n+2}. Then the object penalty is equal to $D(n+3+U)$; the object o_1 is not served in the direct voyage, and time consuming service of the ordinary objects under the given circumstances equals to $U - \varepsilon$ time, where ε is a number from the set $\{0, 1, \dots, U\}$. The fact that servicing of the object o_1 is accomplished at the moment $T = 2n + 3U + 5 + \varepsilon$ is easily calculated for the case. The total penalty for all objects involved equals to $D((n+3) + U) + T$.

It should be noted that $T \leq 2n + 4U + 5$ and we assume that $D = 2n + 4U + 6$. The values thus prescribed prompt that the total penalty for all the objects does not exceed $D(n+3+U+1)$; this is true in case the object o_1 is being served in the direct voyage or the time exceeding U is consumed for ordinary object service in the direct voyage; otherwise, the object o_1 is being served in the return voyage, and the time not exceeding U is consumed for the ordinary object service in the direct voyage, then this penalty is less than $D(n+3+U+1)$.

Service problem thus constructed where total penalty criteria to be minimized, reveals that the object o_1 is to be served in the return voyage and the time $U - \varepsilon$ is thought to be consumed for the ordinary object service in the direct voyage. Total penalty for all objects becomes equal to $D(n+3+U) + 2n + 3U + 5 + \varepsilon$. Lower estimate for the total penalty is the value $D(n+3+U) + 2n + 3U + 5$ obtained when $\varepsilon = 0$. We are in a position to obtain this estimate provided that there is a subset of the ordinary objects with the total servicing duration U, i.e. if the initial partition problem has a positive answer.

The above model is mathematically reduced in the polynomially dependent time of input data available. The theorem has thus been proved.

6. Conclusion

Here we have presented a mathematical model of servicing where a moving processor is to serve a set of the stationary objects positioned within the one-dimensional working zone. We have defined the optimization problems with one and two criteria to evaluate the quality of servicing schedules for the set of the objects. These criteria correspond to situations arising in operative management of fuel supply to diesel-electrical complexes afloat or dredgers.

We have developed the algorithms based on the dynamic programming and the Pareto concept; the implementations of these algorithms have been demonstrated on numerical examples.

We have shown that the algorithm for the problem of processing time minimization is polynomial, and that the problem of total penalty minimization is NP-hard. Correspondingly, the bi-criteria problem with the mentioned evaluation criteria is fundamentally intractable, computational complexity of the schedule structure algorithm is exponential.

The entailing circumstances considered are not practically critical, since the number of stationary objects does not exceed 13–15 units in existent production systems allowing for the construction of an optimal schedule within an hour. In the meantime, there may essentially exist larger amount of objects in production systems other than stated in view of the considered mathematical model, and, as the case may be, it will call for efficient heuristics approaches (see [22–25]) to seek optimization solutions.

References

[1] Kogan D. I., Fedosenko Yu. S., "Zadacha dispetcherizatsii: analiz vychislitel'noy slozhnosti i polinomial'no razreshimye podklassy", *Diskretnaya matematika RAN*, **8**:3 (1996), 135–147, [in Russian].

[2] Tanaev V. S., Gordon V. S., Shafransky Y. N., *Scheduling Theory*, Single-Stage Systems, Springer Science+Business Media, 1994.

[3] T'kindt V., Billaut J., *Multicriteria scheduling: models and algorithms*, Springer, 2006.

[4] Brucker P., Knust S., *Complex scheduling*, Springer, 2006.

[5] Brucker P., *Scheduling Algorithms*, Springer, 2007.

[6] Pinedo M., *Scheduling: Theory, Algorithms, and Systems*, Springer, 2008.

[7] Shen H., *Optimal Scheduling for Satellite Refueling in Circular Orbits*, PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2003.

[8] Klimin A. V., Poedinok V. M., "Dozapravka v polete grazhdanskikh samoletov: perspektivy i problemy", *TsAGI*, http://www.tsagi.ru/cgi-bin/jet/viewnews.cgi?id=20101230080777618957, [in Russian].

[9] Yu P., *Multiple Criteria Decision Making: Concepts, Techniques and Extensions*, Plenum Press, NY, 1985.

[10] Steuer R. E., *Multiple Criteria Optimization: Theory, Computation and Application*, J.Wiley&Sons Inc., NY-Chichester-Brisbane-Toronto-Singapore, 1986.

[11] Ehrgott M., *Multicriteria Optimization*, second ed., Springer, 2005.

[12] Podinovskiy V. V., Nogin V. D., *Pareto-optimal'nye resheniya mnogokriterial'nykh zadach*, Fizmatlit, 2007, [in Russian].

[13] Bellman R., Dreyfus S. E., *Applied dynamic programming*, Princeton, 1962.

[14] Sigal I. X., Ivanova A. P., *Vvedenie v prikladnoe diskretnoe programmirovanie: modeli i vychislitel'nye algoritmy*, Fizmatlit, 2007, [in Russian].

[15] Garey M. R., Johnson D. S., *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman & Co., 1979.

[16] Arora S., Barak B., *Computational Complexity: A Modern Approach*, Princeton, 2009.

[17] Lipton R. J., *The P=NP Question and Gödel's Lost Letter*, Springer Science+Business Media, 2010.

[18] Villareal B., Karwan M., "Multicriteria Dynamic Programming with an Application to the Integer Case", *Journal of optimization theory and applications*, **38**:1 (1982), 43–69.

[19] Kogan D. I., *Dinamicheskoe programmirovanie i diskretnaya mnogokriterial'naya optimizatsiya*, Izd-vo NNGU, N. Novgorod, 2004, [in Russian].

[20] Dunichkina N. A., Kogan D. I., Pushkin A. M., Fedosenko Yu. S., "Ob odnoy modeli obsluzhivaniya statsionarnykh ob"ektov peremeshchayushchimsya v odnomernoy rabochey zone protsessorom", *Trudy*, XII vserossiyskoe soveshchanie po problemam upravleniya VSPU-2014 (Moskva, 16-19 iyunya 2014g.), Institut Problem Upravleniya im. V.A.Trapeznikova RAN, 2014, 5044–5052, [in Russian].

[21] Kogan D. I., Fedosenko Yu. S., "Optimal servicing strategy design problems for stationary objects in a one-dimensional working zone of a processor", *Automation and remote control*, **71**:10 (2010), 2058–2069.

[22] Dorigo M., *Optimization, Learning and Natural Algorithms*, PhD thesis, Dipartamento di Elettronica, Politecnico di Milano, 1992.

[23] Blum C., Roli A., "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", *ACM Computing Surveys*, **35(3)** (2003), 268–308.

[24] Jones M. T., *Al Application Programming*, Charles river media, Inc., Hingham, Massachusetts, 2003.

[25] Dunichkina N., "Algorithms for bi-criteria single-machine scheduling problem of servicing a spaced group of objects", *Proceedings of the 3th International IT Conference "Information Technology in modern life", in: Abstracts of presentations*, Bonn-Rhein-Sieg University of Applied Sciences, 2008, 4.

Построение расписаний обслуживания стационарных объектов перемещающимся в одномерной зоне процессором

Дуничкина Н. А., Коган Д. И., Федосенко Ю. С.

Волжский государственный университет водного транспорта
603005 Россия, г. Нижний Новгород, ул. Нестерова, 5а
Московский государственный университет приборостроения и информатики
107996 Россия, г. Москва, ул. Стромынка, 20

Ключевые слова: теория расписаний, динамическое программирование, принцип Парето, NP-трудность, многокритериальная оптимизация

Рассматривается математическая модель, в которой мобильный процессор, перемещаясь в пределах одномерной рабочей зоны, реализует однофазное однократное обслуживание рассредоточенной в пределах этой зоны совокупности стационарных объектов. В процессе перемещений в рабочей зоне процессор совершает два рейса – прямой и обратный. При этом часть объектов обслуживается в прямом рейсе, остальные объекты – в обратном рейсе. Обслуживание любого объекта нельзя начать ранее предписанного ему срока. С каждым объектом ассоциирован индивидуальный штраф, являющийся монотонно возрастающей функцией от момента завершения его обслуживания. В качестве минимизируемых критериев оценки качества расписаний обслуживания выступают момент завершения работ по всей совокупности объектов и величина суммарного штрафа по ним. Ставятся и исследуются оптимизационные задачи с одним и двумя критериями оценки, конструируемые решающие алгоритмы основаны на принципе динамического программирования и концепции Парето; последовательная их реализация продемонстрирована на численных примерах. Показано, что алгоритм решения задачи на оптимальное быстродействие является полиномиальным, а задача построения расписания обслуживания, обеспечивающего минимизацию величины суммарного штрафа по всем объектам, является NP–трудной. Соответственно бикритериальная задача с указанными критериями оценки относится к числу труднорешаемых, вычислительная сложность алгоритма построения расписания обслуживания является экспоненциальной. Модель описывает процессы снабжения топливом плавучих дизель-электрических комплексов, осуществляющих русловую добычу инертных строительных материалов в крупномасштабных районах речных путей. Модели и оптимизационные задачи, подобные рассматриваемым, представляют интерес для таких приложений, как управление дозаправкой топливом орбитальной группировки спутников и магистральных гражданских самолетов.

Статья публикуется в авторской редакции.

Сведения об авторах:
Дуничкина Надежда Александровна,
Волжский государственный университет водного транспорта,
кандидат физ.-мат. наук, старший научный сотрудник,
ORCID 0000-0002-4347-5116
Коган Дмитрий Израилевич,
Московский государственный университет приборостроения и информатики,
доктор техн. наук, профессор,
ORCID 0000-0002-1203-1539
Федосенко Юрий Семенович,
Волжский государственный университет водного транспорта,
доктор техн. наук, профессор,
ORCID 0000-0001-5582-2325