Федеральное агентство морского и речного транспорта
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волжский государственный университет водного транспорта»

Кафедра иностранных языков и конвенционной подготовки

РЕФЕРАТ
по английскому языку

THE SYNTHESIS OF SERVICE POLICIES OF OBJECTS FLOW
IN THE SYSTEM WITH REFILLABLE STORAGE COMPONENT

СИНТЕЗ СТРАТЕГИЙ ОБСЛУЖИВАНИЯ ПОТОКА ОБЪЕКТОВ
В СИСТЕМЕ С НАКОПИТЕЛЬНО-РАСХОДНЫМ ЭЛЕМЕНТОМ

Выполнил: А.С. Пудов

Проверил: Л.Г. Орлова
доцент, к.фил.н.

Нижний Новгород
2017

CONTENT

ABSTRACT

This paper describes the problem that appears in research that aims at implementing a fuel management system for inhabitants of the Northern Territories who are interesting in bulk water-transport delivery. The mathematical model of a logical scheme is constructing in form of uniprocessor system with refillable storage element that carries out single-stage service of deterministic objects flow. We formulate the task scheduling optimization problem in the form where service policies are estimates by two independent minimized criteria. Because of our research, we also provide synthesis algorithm of Pareto-efficient service policies that utilizes bicriteria approach of dynamic programming. The feasibility of the algorithm is demonstrated by numerous examples and results derived from computational experiments.

INTRODUCTION

Diesel is the main energy source used for life support and productive infrastructure in the Arctic regions. Bulk fuel delivery is carried out by water transport during a short navigation period. Typical petroleum delivery logistics should consider navigation in the Arctic regions, looks as follows.

Inland-waterway tankers with a capacity of 3 to 5 thousand tons deliver diesel fuel to the base oil ports, where the fuel is pumped into the system of interconnected onshore tanks for intermediate storage. Later, the fuel is taken from the intermediate storages by shallow-draft tankers with a capacity of 600 to 800 tons and delivered by small rivers to the points of consumption. Unloading and loading of diesel fuel carried out at a specialized oil port terminal.

Described logistics uses tankers, which are characterized by a variety of technical and economic parameters. The key objective of the scheduling problem is to develop an effective cargo handling management strategy for a specialized oil port terminal. As a rule, an effective strategy, must meet the following requirements:

a) minimizing the total costs caused by unproductive downtime fleet;

b) reducing downtime over the required values. However, other criteria for evaluating the effectiveness of management strategies can be significant, depending on the regional particularities and the evolving operational environment.

The number of possible strategies is exponentially depending on the number of vessels and does not depend on the criteria [1, 2]. This means that even with a relatively small number of vessels, synthesis of the most efficient strategy can be carried out only in the information system, equipped with a specialized sufficiently fast solving algorithms. Mass delivery of oil in the Arctic region during spring and summer, a period between the reception of information on the incoming fleet units and the beginning of the first ship's processing is relatively small. This period is usually no more than 20-30 minutes, in the certain cases. Therefore, the key task of the base oil port controller is to define (later to implement) an effective strategy for cargo tankers processing.

The objective of this research work is to develop an algorithm that satisfies specified requirements (a and b bullets) and provides a synthesis of strategies for almost a reasonable period of time. In order to achieve this goal the following items must be met:

– construct a mathematical model of a single-processor system with refillable storage component that satisfies the above logistics scheme;

– define a synthesis of an optimal service management strategy problem;

– design an optimal synthesis problem-solving algorithm using a dynamic programming approach [3];

– implement an algorithm for a model example and show execution results of computational experiments to assess its performance.

MATHEMATICAL MODEL

Each object of finite stream O_k ={o(1), o(2), …, o(k)} is subject to single-phase servicing by a stationary processor P with a refillable storage component. The stream O_k consists to two O^+ and O^- such, that $O^+ \cup O^- = O_k$ and $O^+ \cap O^- = \varnothing$. Objects of substream O^+ are intended to fill the reservoir; and objects of substream O^- to be filled from reservoir. The collection of object indices from substreams O^+, O^- are declared as Q^+ and Q^-.

For each object $o(i)$, $i = \overline{1,k}$, the following integer values are known:

t_i – the moment when this object arrives at the servicing queue P,

τ_i – the normal duration of servicing,

v_i – the volume characteristic,

a_i – the penalty per unit of time that the object spends in the servicing system P,

d_i – the directive deadline of servicing ($d_i \geq \tau_i$),

w_i – the binary parameter, indicating which substream consists the object to. In such way w_i = +1, if $i \in \mathrm{Q}^+$ and w_i = -1, if $i \in \mathrm{Q}^-$.

Objects have indices in the order of their arrival in P, i.e. $0 \leq t_1 \leq t_i \leq \ldots \leq t_k$. The refillable storage component represented by a reservoir of capacity V^*. At moment t the value of the filling is characterized by a variable V_t, and thus V_0 – is the value of the filling at the initial moment $t = 0$. As a result of servicing an object of substream O^+ (O^-), the reservoir filling increases (decreases) by v_i value.

We assume that the processor P is ready to service objects of the stream O_k starting from time moment $t = 0$. Following restrictions imposed on servicing:

− servicing each object done without interruption;

− unserved object cannot leave the queue;

− the processor cannot service more than one object at a time;

− the processor has no idle time.

A service scheduling S for the stream O_k is defined as a permutation $S = \{i_1, i_2, \ldots, i_k\}$ of the collection of indices $N = \{1, 2, \ldots, k\}$ [4]; when it is implemented, an object with index i_j as the j-th in the queue $(j = \overline{1,k})$.

Obviously, starting from moment t, servicing an object $o(s)$ of substream O^+ $(s \in Q^+)$, is only possible if the reservoir has sufficient capacity, i.e. when condition $V_t + v_s \leq V^*$ is true. Similarly, servicing an object $o(s)$ of substream $O^-(s \in Q^-)$ is only possible when following condition $V_t - v_s \geq 0$ is true. Let's call the system of inequalities capacity restrictions, and in case they are true for all objects $o(i_j)$, $j = \overline{1,k}$, of strategy S, then consider them valid. Let Ω denote the collection of admissible servicing strategies.

Next, we will always assume the following relations

$$2 \cdot v_i \leq V^* \ (i = \overline{1,k}).$$

Then necessary and sufficient condition for the non-emptiness of the set of admissible strategies Ω defined as the system of inequalities

$$0 \leq V_0 + \sum_{i=1}^{n} w_i \cdot v_i \leq V^*.$$

OPTIMIZATION PROBLEM

For an arbitrary object o(ij), each admissible servicing strategy S = {i1, i2, …, ik} unambiguously defines the time moments when its servicing begins and ends for; in what follows we will denote these moments $t^*(i_j, S)$ and $\bar{t}(i_j, S), j = \overline{1,k}$ respectively.

When implementing a strategy S, the total penalty $K_1(S)$ over all objects from the substream O^+ is $\sum_{i_j \in Q^+} a_{i_j}(\bar{t}(i_j, S) - t_{i_j})$, and criterion $K_2(S) = \max_{i_j \in Q^-}(\max(\bar{t}(i_j, S) - d_{i_j}, 0))$ determines the maximal value for violating directive deadline in the realization of servicing among all objects from substream O^-. The bicriterial problem that we will study below formulated as follows.

Find the complete collection of Pareto [5] effective estimates in the problem of minimizing a total penalty $K_1(S)$ for all objects of substream O^+ and in the problem of maximizing a total penalty $K_2(S)$ for all objects of stream O^-:

$$\{\min_{S \in \Omega}(K_1(S)), \min_{S \in \Omega}(K_2(S))\}.$$

1)

In this approach, the methodology to determine the servicing strategy provides the consistent implementation of the following steps:

1) synthesize the complete collection of effective estimates in bicriterial problem (1);

2) decision maker have to select an effective estimate to realize, based on results of Step 1;

3) construct the Pareto optimal solution that generates efficient estimates chosen on Step 2.

Without providing a proof, we note that the problem 1* is NP-hard [6]. The corresponding solving algorithm for this problem is easy to construct and its implementation listed below.

Algorithm for the bicriterial approach of dynamic programming [7, 8]. We now introduce additional notation needed in what follows. Let Y is the set of two-dimensional vectors, and let $x = (x_1, x_2)$ is a given arbitrary two-dimensional vector. We $Y \otimes x$ denote the collection of all vectors of the form (y1 + x1, max(y2, x2)), where (y1, y2) $\in Y$.

We say that vector (x'_1, x'_2) dominates vector (x_1, x_2), if conditions $x'_1 \leq x_1$ and $x'_2 \leq x_2$ are true; wherein at least one of these inequalities is satisfied as a strict. For an arbitrary set of estimate vector X, we $eff[X]$ denote subset of the elements belonging to X and non-dominating it.

Let $F(t)$ is the subset of object indices from the stream O_k, that arrives at servicing at moment t. We denote the collection of object indices for object arrives over interval $[t + 1, t + \Delta]$, $\Delta \geq 1$ by $D(t, \Delta)$. It is obvious, that

$$D(t,\Delta) = \bigcup\nolimits_{g=1}^{\Delta} F(t+g).$$

During the servicing of stream objects O_k, the decision made at those moments when the system is free and we must choose another object from one of the substreams. Thus, it is necessary to take into account the current filling of the reservoir. Accordingly, (t, Q, V_t) is the current state of servicing at moment t, where Q – is the set of indices for objects of stream O_k, waiting for service at moment t.

At any state of servicing (t, Q, V_t), $t < t_k$ the set of unserved objects Q will always additionally include a dummy (zero) object $o(0)$ with characteristics $a_0 = 0$, $\tau_0 = 1$, $v_0 = 0$, $d_0 = 0$ and to be definite, we let $w_0 = +1$. Servicing a dummy object means processor downtime during one processing cycle.

Processor downtime occurs in following cases:

− all previously arrived objects have already been serviced, and other objects from the stream have not yet arrived;

− none of the objects that have previously arrived and are awaiting servicing from time moment t can be accepted for servicing because the reservoir either does not have enough free space (for objects from substream O^{+}) or does not have enough product (for objects from substream O^{-});

− it makes sense to let the processor go idle in order to ensure high priority servicing of some object with high penalty per unit of idle time that has not yet arrived.

The idle object is excluded from set Q when $t \geq t_k$, i.e., when all objects of stream O_k have arrived and awaiting servicing.

$E(t, Q, V_t)$ is the collection of effective two-dimensional estimates, constructed by servicing all objects with indices of set Q and objects that will arrive to servicing after the moment t. Then $E(0, F(0), V_0)$ is the complete collection of effective estimates for the initial bicriterial problem.

In current state (t, Q, V_t) object $o(s)$ $s \in Q$ can be served only if the following condition is true $0 \le V_t + w_s v_s \le V^*$. Q^* is a set of object indices that can be served at state (t, Q, V_t), $Q^* \subset Q$.

Obviously, that for any $\theta \ge 0$ и $Q = \{\alpha\}$, where α is the index of the arbitrary object, the following condition defined:

$$E(t_k + \theta, \{\alpha\}, V_{t_k + \theta}) = \{a_\alpha(t_k + \theta + \tau_\alpha - t_\alpha), \max(\bar{t}(\alpha, S) - d_\alpha, 0)\}. \qquad 2)$$

In state (t, Q, V_t), if an object with index $\alpha \in Q^*$ chosen for servicing, then $t + \tau_\alpha$ is the next decision-making moment, and the choice of an index of the next object for servicing has to be from a set $(Q \setminus \{\alpha\}) \bigcup D(t, \tau_\alpha)$. Characteristic of refillable storage component will change to $V_t + w_\alpha v_\alpha$. Summing up, we have that

$$E(t, Q) = eff[\bigcup_{\alpha \in Q^*} [(a_\alpha(t + \tau_\alpha - t_\alpha), \max(\bar{t}(\alpha, S) - d_\alpha, 0)) \otimes \\ \otimes E(t + \tau_\alpha, (Q \setminus \{\alpha\}) \bigcup D(t, \tau_\alpha)), V_t + w_\alpha v_\alpha)]. \qquad 3)$$

Formulas (2) and (3) are recurrent dynamic programming relations for solving the bicriterial problem (1).

The computational complexity of described algorithm is characterized by

$$O(L \cdot 2^n \cdot n \cdot \max_{i: o_i \in O^-}(\max(L - d_i, 0))), \text{ where } L = t_n + \sum_{i=1}^{n} \tau_i .$$

The following example illustrates implementation of described algorithm.

It is required to find the complete collection of Pareto effective estimates and related Pareto-optimal servicing strategies for objects of stream O_4. $V^* = 19$ is the maximum capacity of the reservoir, at initial moment $t = 0$ and filling $V_0 = 10$. The model data set is listed below.

Table 1. Model data set.

№	t_i	τ_i	a_i	d_i	v_i	w_i
1	0	3	2	5	5	1
2	2	4	3	7	6	1
3	4	2	5	9	2	-1
4	5	4	6	9	7	-1

Complete collection of efficient estimates is the set $E(0, \{0, 1\}, 10)$. Let us say, in set Q, let the index of dummy object is 0 and set an index 0 to state (0, {0, 1}, 10). By formula (3) we have

$$E(0, \{0, 1\}, 10) = \mathit{eff}[(0, 0) \otimes E(2, \{0, 1, 2\}, 10), (0, 0) \otimes E(3, \{0, 2\}, 15)].$$

Set indices 1 and 2 to states (2, {0, 1, 2}, 10) и (3, {0, 2}, 15) respectively. During the algorithm execution, each new state of servicing system receives the next index.

By formulas (2) and (3), we find recursively sets of two-dimensional estimates $E(2, \{0, 1, 2\}, 10)$ and $E(3, \{0, 2\}, 15)$.

In order to calculate $E(2, \{0, 1, 2\}, 10)$ is no point to choose a dummy object, because servicing an object with index 1 can be completed before arrival of object with index 3. In addition, it is impractical to servicing an object with index 1, because servicing can start earlier that moment $t = 2$. Therefore, $E(2, \{0, 1, 2\}, 10) = \mathit{eff}[3(2 - 2), \max(6 - 7, 0)) \otimes E(6, \{1, 3, 4\}, 16)] = E(6, \{1, 3, 4\}, 16)$.. Since all objects are entered into servicing system in state (6, {1, 3, 4}, 16), then, starting from this step, we are excluding dummy object from consideration. By formula (3) we get $E(6, \{1, 3, 4\}, 16) = \mathit{eff}[(2(6 - 0), \max(9 - 5, 0)) \otimes E(9, \{3, 4\}, 21); (5(6 - 4), \max(8 - 9, 0)) \otimes E(8, \{1, 4\}, 14); (6(6 - 5), \max(10 - 9, 0)) \otimes E(10, \{1, 3\}, 9)] = \mathit{eff}[(12, 4) \otimes E(9, \{3, 4\}, 21); (10, 0) \otimes E(8, \{1, 4\}, 14); (6, 1) \otimes E(10, \{1, 3\}, 9)]$.

If serve the object with index 1 in state $E(6, \{1, 3, 4\}, 16)$, filling the reservoir exceed the maximum possible value $V^* = 19$. Thus, to find $E(6, \{1, 3, 4\}, 16)$, we need to find $E(8, \{1, 4\}, 14)$ и $E(10, \{1, 3\}, 9)$. $E(8, \{1, 4\}, 14) = \mathit{eff}[(16, 6) \otimes E(11, \{4\}, 19); (18, 3) \otimes E(12, \{1\}, 7)]$. According to formula (3), we have $E(11, \{4\}, 19) = (36, 6)$, and $E(12, \{1\}, 7) = (24, 10)$.

Similarly, we get $E(10, \{1, 3\}, 9) = \{(65, 8), (54, 10)\}$ and begin to compute $E(6, \{1, 3, 4\},16) = \{(62, 6), (52, 10)\}$. As $E(2, \{0, 1, 2\}, 10) = E(6, \{1, 3, 4\})$, then $E(2, \{0, 1, 2\}, 10) = \{(62, 6), (52, 10)\}$.

Arguing similarly, we find $E(3, \{0, 2\}, 15) = \{(42, 5), (30,7)\}$. As a result, we get complete collection of efficient estimates $E(0, \{0, 1\}, 10) = \mathit{eff}[E(2, \{0, 1, 2\}, 10), E(3, \{0, 2\}, 15)] = \mathit{eff}[(62,$

6), (52, 10), (42, 5), (30, 7)] = {(42, 5), (30, 7)}.

Let estimate (42, 5) chosen from the collection of estimates. Then it is easy to prove that corresponding Pareto-optimal strategy servicing strategy has the form {1, 3, 2, 4}.

Similarly, we can find {1, 3, 4, 2} Pareto optimal servicing strategy on the estimate (30, 7).

Experimental results. Development is done using PC with Intel Core 2 Duo, 3.16GHz CPU and 4Gb memory. The algorithm is implemented in C++ and compiled with Microsoft Visual Studio compiler.

Calculations made for stream O_k parameters from the following range:

$$t_{i-1} \le t_i \le t_{i-1} + 5,\ 1 \le a_i \le 11,\ 1 \le \tau_i \le 15,\ t_i + \tau_i \le d_i \le t_i + \tau_i + 5,\ 1 \le v_i \le V^*/2,\ i = \overline{1, k}.$$

The size of the stream k has changed from 8 to 15 with a single step. The volume of the reservoir chosen in the range $100 \le V^* \le 300$.

Chosen bounds of data set changes complies with the transport applications of this type and describes the real balance of parameters.

For a fixed value of parameter k, according to uniform distribution law, generated a series of 10^3 individual problems. In each problem, we choose a complete collection of Pareto effective estimates and related strategy sets.

Experimental results are shown in Table 2, where t_{avg} – average time, t_{max} – maximum time, and t_{min} – minimum time to solve a problem.

Table 2. Experimental results.

k	t_{avg}, c	t_{max}, c	t_{min}, c	k	t_{avg}, c	t_{max}, c	t_{min}, c
8	0.001	0.016	0.000	12	0.871	24.063	0.000
9	0.003	0.031	0.000	13	5.006	73.236	0.000
10	0.020	0.297	0.000	14	45.350	633.016	0.000
11	0.112	1.074	0.000	15	447.542	2344.020	0.000

As shown in Table 2, the dynamic programming algorithm, although characterized by exponential computational complexity, can perform the synthesis of the Pareto-optimal set of object flow servicing strategies for practically significant dimension values for the period of time that does not exceed an average of 10 minutes. The time intervals are acceptable for the synthesis of objects service strategies in logistics system of this type.

CONCLUSION

This paper gives a complete description of the logical scheme used, in particular, for a diesel delivery to consumers in the Arctic region. Built a mathematical model satisfies this scheme. For this model, formulated a bicriterial optimization problem and designed its solving algorithms, based on Pareto concept and a multi-criteria discrete dynamic programming approach. The implementation of the algorithm demonstrated by an example. Based on the massive computational experiments of software implementation for the algorithm, the possibility of its regular use in the transport and process control system has been proved.

In solving the synthesis of servicing strategies with increased dimensions problem, the actual problem is to constructing such modifications of management model, which generates subclasses of optimization problems that can be solved in polynomial time, along with fulfilling the conditions of application usage.

REFERENCES

[1] Kogan, D.I. and Fedosenko, Yu.S. The discretization problem: analysis of computational complexity and polynomially solvable subclasses. Discret. Math. Appl., 1996, vol. 6, Issue 5, pp. 435–447. DOI:10.4213/dm534

[2] Kogan, D.I., Kuimova, A. S., Fedosenko, Yu.S. The problems of servicing of the binary object flow in system with refillable storage component. Automation and Remote Control, 2014, vol. 75, Issue 7, pp. 1257–1266. DOI:10.1134/S0005117914070078

[3] Bellman, R.E. and Dreyfus, S.E. Applied Dynamic Programming. – Princeton: Princeton Univ. Press, 1962, p. 390.

[4] Tanaev V.S., Sotskov, Y.N., Strusevich, V.A. Scheduling Theory: Multi-Stage Systems. Springer Netherlands, 2012. – 406 p.

[5] Pinedo M.L. Planning and Scheduling in Manufacturing and Services. Springer, 2009, 537 p.

[6] Garey, M.R. and Johnson, D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, 1990. – 338 p.

[7] Villareal, B. and Karwan, M.H Multicriterial Dynamic Programming with an Application to the Integer Case. Journal of optimization theory and applications, 1982, vol. 38, Issue 1, pp. 43–69. DOI:10.1007/BF00934322

[8] Kogan D.I., Fedosenko, Yu.S., Dunichkina N.A. Bicriterial servicing problems for stationary objects in a one-dimensional working zone of a processor. Automation and Remote Control, 2012, vol. 73, Issue 10, pp. 1667–1679. DOI:10.1134/S0005117912100074