МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

УДК 519.8

ЗАДАЧА ОПТИМИЗАЦИИ СТРАТЕГИИ ОБСЛУЖИВАНИЯ БИНАРНОГО ПОТОКА ОБЪЕКТОВ ДВУМЯ MOBILE-ПРОЦЕССОРАМИ В ЛИНЕЙНОЙ РАБОЧЕЙ ЗОНЕ

М.Б. Резников, Ю.С. Федосенко

Волжская государственная академия водного транспорта

mikerez@mail.ru

Поступила в редакцию 31.05.2007

Рассматривается задача синтеза оптимальной стратегии однофазного обслуживания конечного детерминированного потока объектов двумя mobile-процессорами в одномерной рабочей зоне. В качестве критерия оптимизации выступает доход за обслуживание объектов потока. Сформулирована математическая модель обслуживания, разработан алгоритм синтеза оптимальной стратегии обслуживания и выполнена оценка его вычислительной сложности. Приводятся результаты вычислительных экспериментов.

Введение

Решается задача синтеза оптимальной стратегии обслуживания объектов конечного детерминированного потока при транзитном прохождении ими существенно протяженной рабочей зоны двух mobile-процессоров [1]. Поток объектов обладает свойством бинарности, т.е. состоит из двух подпотоков: объекты одного подпотока поступают в рабочую зону через ее «левую» граничную точку, а объекты другого подпотока – через «правую» точку. Внутри рабочей зоны объекты совершают независимое равномерное поступательное движение. При этом для каждого объекта известны моменты времени поступления и выхода из рабочей зоны, а также длительности и доходы за обслуживание.

Принципиальным отличием рассматриваемой модели от ранее исследованной [2] является наличие в рабочей зоне не одного, а двух независимых и не взаимно заменяемых обслуживающих процессоров, точки стартового расположения которых известны. Каждый из процессоров может находиться в рабочей зоне в одном из трех состояний: поступательное равномерное движение со «своей» постоянной скоростью; обслуживание объекта потока в процессе совместного с ним поступательного движения со скоростью последнего; простой. Любой объект потока в процессе транзитного прохождения им рабочей зоны может быть обслужен обоими процессорами либо только одним (любым) процессором или остаться необслуженным; одновременное обслуживание одного объекта обоими процессорами запрещено.

Задача заключается в построении стратегии обслуживания, обеспечивающей максимизацию суммарного дохода по всем обслуженным объектам потока.

На содержательном уровне рассматриваемая модель описывает сравнительно новую для районов интенсивного речного судоходства технологию технического обслуживания транспортных единиц, транзитно проходящих зону ответственности сервисного предприятия. Обслуживание осуществляется «на ходу» двумя независимыми специализированными самоходными судами, каждое из которых предназначено для выполнения только одного вида работ. Основная задача диспетчера сервисного предприятия заключается в реализации такой стратегии обслуживания нуждающихся в технических услугах транзитных судов, которая обеспечивает получение максимальной суммарной прибыли.

При этом существенным обстоятельством, оказывающим влияние на работу диспетчера, являются весьма жесткие ограничения на допустимую длительность штатного регламента формирования стратегии обслуживания путем решения соответствующих экстремальных задач в (on-line)- или (off-line)-режимах. В силу сказанного актуальной является проблема разработки достаточно быстрых (по условиям конкретного применения) алгоритмов синтеза стратегий обслуживания.

Постановка задачи

Имеется n-элементный детерминированный поток независимых объектов $O = \{o(1), o(2), \ldots, o(n)\}$. Каждый объект $o(i)$ ($i=\overline{1,n}$) проходит транзитом с постоянной скоростью общую рабочую зону Ξ двух независимых и не взаимно заменяемых mobile-процессоров P^1 и P^2. Зона Ξ представляет собой отрезок AB длиной L. В пределах рабочей зоны процессоры могут перемещаться как в автономном режиме с постоянной скоростью (соответственно $u^1{}_A$, $u^2{}_A$ – в направлении от A к B и $u^1{}_B$, $u^2{}_B$ – в противоположном направлении), так и в паре с обслуживаемым объектом со скоростью последнего. Осуществляемое процессором $P^1(P^2)$ обслуживание однофазное без прерываний. Одновременное обслуживание процессором более одного объекта, а также одновременное обслуживание одного объекта двумя процессорами запрещены. Освободившийся от обслуживания очередного объекта процессор считается готовым к обслуживанию следующего. Поток $O(n)$ обладает свойством бинарности, т.е. состоит из двух подпотоков O_A и O_B таких, что $O_A \cup O_B = O$ и $O_A \cap O_B = \varnothing$. Объекты подпотока O_A входят в зону Ξ через граничную точку A и проходят ее поступательно в направлении к точке B, а объекты подпотока O_B поступают в Ξ через граничную точку B и проходят зону Ξ в противоположном направлении. Объекты подпотока O_A покидают рабочую зону через точку B, а объекты подпотока O_B – через точку A.

Для дальнейшего удобно считать отрезок AB разбитым на целое число l элементарных участков длины $\lambda = L/l$. Очевидно, что с необходимой точностью целочисленность λ всегда может быть обеспечена за счет выбора единицы измерения длины L. Соответственно этому разбиению элементарные участки последовательно пронумерованы числами 1, 2, …, l–1, l в направлении от A к B. При этом точка A входит в элементарный участок с номером 1, а точка B – в участок с номером l.

Каждый объект $o(i)$ ($i = \overline{1,n}$) характеризуется следующими целочисленными параметрами:

$t(i)$ – момент поступления в зону Ξ ($0 < t(1) < < t(2) < \ldots < t(n)$);

$\tau^1(i)$, $\tau^2(i)$ – нормы длительности обслуживания объектов процессорами P^1 и P^2 ($\tau^1(i) > 0$, $\tau^2(i) > 0$);

$u(i)$ – указатель принадлежности подпотоку O_A или O_B ($u(i) = 1$, если $o(i) \in O_A$, и $u(i) = 0$, если $o(i) \in O_B$);

$v(i)$ – скорость движения объекта, которая измеряется числом проходимых в единицу времени элементарных участков; целочисленность $v(i)$ также может быть обеспечена выбором соответствующей единицы измерения;

$w^1(i)$, $w^2(i)$ – величина дохода за обслуживание объекта процессорами P^1 и P^2 ($w^1(i) > 0$, $w^2(i) > 0$).

Далее будем считать, что $u^1{}_A, u^2{}_A, u^1{}_B, u^2{}_B < < v(i)$ ($i = \overline{1,n}$), т. е. скорости процессоров меньше скорости самого медленного объекта. Также примем, что взаимные обгоны объектов на отрезке AB исключены.

Возможны следующие автономные состояния процессоров P^1 и P^2, находящихся в начальный момент времени $t = 0$ на элементарных участках x^1 и x^2 зоны Ξ соответственно ($x^1, x^2 \in \in \{1, 2, \ldots, l\}$):

а) движение в направлении от A к B со скоростью $u^1{}_A$, $u^2{}_A$;

б) движение в направлении от B к A со скоростью $u^1{}_B$, $u^2{}_B$;

в) простой.

Стратегию r^1 работы процессора P^1 определяем как кортеж $\{\langle j^1{}_1, j^1{}_2, \ldots, j^1{}_{k(1)}\rangle, \langle \sigma^1{}_1, \sigma^1{}_2, \ldots, \sigma^1{}_{k(1)}\rangle\}$, где $j^1{}_1, j^1{}_2, \ldots, j^1{}_{k(1)}$ – последовательность номеров объектов в порядке их обслуживания, $\sigma^1{}_1, \sigma^1{}_2, \ldots, \sigma^1{}_{k(1)}$ – номера элементарных участков начала обслуживания объекта $j^1{}_g$ ($g = \overline{1,k(1)}$), а $k(1)$ – общее число объектов в стратегии обслуживания потока O процессором P^1. Объекты, не входящие в последовательность $j^1{}_1, j^1{}_2, \ldots, j^1{}_{k(1)}$, обслуживанию процессором P^1 не подлежат.

Стратегию r^2 работы процессора P^2 определим аналогично: $\{\langle j^2{}_1, j^2{}_2, \ldots, j^2{}_{k(2)}\rangle, \langle \sigma^2{}_1, \sigma^2{}_2, \ldots, \sigma^2{}_{k(2)}\rangle\}$; $k(2)$ – общее число объектов в стратегии процессора P^2. Очевидно, что $k(1)+k(2) < 2n$. Общую стратегию r определим как совокупность $\{r^1, r^2\}$.

Для описания фаз-состояний объекта j^m_g ($g = \overline{1,k(m)}$) и процессора P^m, взаимодействующих согласно стратегии ρ^m ($m \in [1, 2]$), введем следующие обозначения:

$\eta(j^m_g, \rho^m)$ – момент начала обслуживания объекта, т.е. момент поступления объекта в начальную точку отрезка σ^m_g;

$\psi(j^m_g, \rho^m)$ – момент прибытия процессора в начальную точку отрезка σ^m_g;

$\varphi(j^m_g, \rho^m)$ – момент окончания обслуживания объекта;

$\gamma(j^m_g, \rho^m)$ – номер участка, на котором обслуживание объекта j^m_g процессором P^m заканчивается;

$\xi(j^m_g, r^m)$ – момент выхода объекта j^m_g из зоны, в которой его обслуживание процессором P^m еще возможно.

Имеют место следующие равенства, однозначно определяющие значения всех введенных выше характеристик:

$$\eta(j_g^m,\rho^m)=\begin{cases}[\sigma_g^m/\nu(j_g^m)]+t(j_g^m)\\ \text{для } j_g^m\in O_A,\\ [(l-\sigma_g^m)/\nu(j_g^m)]+t(j_g^m)\\ \text{для } j_g^m\in O_B;\end{cases} \qquad (1)$$

$$\psi(j_g^m,\rho^m)=\begin{cases}\varphi(j_{g-1}^m,\rho^m)+[(\gamma(j_{g-1}^m,\rho^m)-\\ -\sigma_g^m)/u_B^m],\\ \text{если } \sigma_g^m\le\gamma(j_{g-1}^m,\rho^m);\\ \varphi(j_{g-1}^m,\rho^m)+[(\sigma_g^m-\\ -\gamma(j_{g-1}^m,\rho^m))/u_A^m],\\ \text{если } \sigma_g^m>\gamma(j_{g-1}^m,\rho^m);\end{cases} \qquad (2)$$

$$j(j_g^m, r^m) = h(j_g^m, r^m) + t^m(i); \qquad (3)$$

$$\gamma(j_g^m,\rho^m)=\begin{cases}\sigma_g^m+\nu(j_g^m)\tau(j_g^m) & \text{для } j_g^m\in O_A,\\ \sigma_g^m-\nu(j_g^m)\tau(j_g^m) & \text{для } j_g^m\in O_B;\end{cases}$$

$$\xi(j_g^m,\rho^m)=t(j_g^m)+[l/\nu(j_g^m)]-\tau^m(i). \qquad (4)$$

Фигурирующие в (1), (2) и (4) квадратные скобки означают операцию округления соответствующего частного до ближайшего целого числа.

Каждая стратегия r^m однозначно определяет расчетную величину соответствующей ей прибыли $W(\rho^m)=\sum_{g=1}^{k(m)} w^m(j_g^m)$.

Общая прибыль $W(r)$ системы за обслуживание в результате реализации стратегии r, очевидно, определяется как сумма $W(\rho^1) + W(\rho^2)$.

Множество всех допустимых стратегий обозначим через Ω. Тогда исследуемая экстремальная задача записывается в виде

$$W_{\max}(\rho^*) = \max_{\rho\in\Omega} W(\rho). \qquad (5)$$

Нетрудно показать, что в рамках изложенной модели управления обслуживанием может быть сформулирована NP-трудная задача синтеза оптимальной в смысле (5) стратегии ρ^*. Для ее решения применим метод динамического программирования.

Вывод соотношений динамического программирования

В соответствии с общей концепцией [3] построение рекуррентных соотношений динамического программирования направлено на формирование вычислительной схемы синтеза стратегии ρ^* как некоторой пошаговой процедуры, на каждом этапе которой определяется номер следующего обслуживаемого объекта; при этом пошаговое продвижение по вычислительной схеме реализуется в обратном направлении – от момента завершения обслуживания объектов потока O к его началу.

Пусть t – момент принятия очередного решения. Момент $t = 0$ и каждый из следующих моментов времени, когда процессор P^m ($m \in [1, 2]$) завершает обслуживание очередного объекта, считается моментом принятия решения. Ситуация в произвольный момент принятия решения характеризуется набором (t, m, p, r, θ), где:

t – число тактов, прошедших от начала процесса;

m – номер свободного по состоянию на данный момент процессора, $m \in [1, 2]$;

p – номер элементарного участка, на котором расположен процессор с номером m;

r – номер объекта, обслуживаемого в данный момент другим процессором – $P^{(3-m)}$;

θ – число тактов дискретного времени, оставшихся до завершения обслуживания процессором $P^{(3-m)}$ объекта с номером r.

Считается, что в особой ситуации, когда свободны оба процессора, r соответствует номеру элементарного участка, на котором расположен процессор P^2, а $m = 1$ и $\theta = 0$. Очевидно, что начальная ситуация относится к числу особых и описывается набором $(0, 1, x_1, x_2, 0)$.

Целеуказание (принимаемое решение) заключается в назначении для свободного по состоянию на данный момент процессора очередного подлежащего обслуживанию объекта a. После указанного назначения процессор немедленно начинает движение в направлении навстречу предписанному объекту, если этот объект в данный момент не обслуживается другим процессором, и в направлении точки завершения обслуживания предписанного объекта другим процессором – в противном случае. В случае если точка встречи процессора с назначенным объектом располагается вне отрезка AB, процессор дожидается объект на границе рабочей зоны Ξ.

Совокупность объектов, любой из которых в момент принятия решения t может быть назначен в качестве очередного обслуживаемого находящемуся на элементарном участке с номером p свободному процессору, обозначим через $U(t, m, p, r, \theta)$.

Согласно принятому выше соглашению о соотношении собственных скоростей движения объектов потока, не обслуживаемый в момент t занятым процессором и не обслуженный ранее свободным процессором объект входит в совокупность $U(t, m, p, r, \theta)$, если по состоянию на данный момент он еще не прошел участок с номером p и начинающееся в точке встречи объекта и процессора обслуживание завершится не позднее момента выхода объекта из рабочей зоны. Обслуживаемый в момент времени t другим процессором объект x включается в совокупность $U(t, m, p, r, \theta)$, если точка, от которой свободный процессор может начать обслуживание этого объекта, такова, что освобождение процессора от объекта произойдет не позднее момента выхода объекта из зоны Ξ. Отметим, что для любого процессора при любых значениях аргументов t и p совокупность $U(t, m, p, r, \theta)$ определяется путем выполнения для каждого не обслуженного рассматриваемым процессором объекта простых арифметических вычислений.

Основываясь на соотношениях (2), (3), определим функции $\varphi_1(\alpha, t, m, p, r, \theta)$ и $\varphi_2(\alpha, t, m, p, r, \theta)$, которые вычисляют момент начала обслуживания следующего назначенного объекта a, исходя из ситуации (t, m, p, r, θ). Функция φ_1 соответствует случаю, когда объект a свободен, а φ_2 – случаю, когда объект a в данный момент обслуживается другим процессором:

$$\varphi_1(\alpha,t,m,p,r,\theta)=\begin{cases}[(p-\mu(\alpha,t))/(u_A^m+\\ +v(\alpha))] \quad \text{для } o(\alpha)\in O_B,\\ [(p-\mu(\alpha,t))/(u_B^m+\\ +v(\alpha))] \quad \text{для } o(\alpha)\in O_A;\end{cases} \tag{6}$$

$$\varphi_2(\alpha,t,p,m,r,\theta)=\begin{cases}[(p-\mu(\alpha,t+\theta))/(u_A^m+\\ +v(\alpha))] \quad \text{для } o(\alpha)\in O_B,\\ [(p-\mu(\alpha,t+\theta))/(u_B^m+\\ +v(\alpha))] \quad \text{для } o(\alpha)\in O_A.\end{cases} \tag{7}$$

Координата $\mu(i, t)$ объекта $o(i)$, входящая в соотношения (6), (7), в момент времени t определяется по формуле

$$\mu(i,t)=\begin{cases}(t-t(i))v(i) & \text{для } o(i)\in O_A,\\ l-(t-t(i))v(i) & \text{для } o(i)\in O_B.\end{cases}$$

Определив момент начала обслуживания следующего назначенного объекта a как

$$\varphi'(\alpha,t,m,p,r,\theta)=\begin{cases}\varphi_1(\alpha,t,m,p,r,\theta), & \text{если } \alpha\neq r,\\ \varphi_2(\alpha,t,m,p,r,\theta), & \text{если } \alpha=r,\end{cases}$$

найдем следующий за t момент принятия решения. При этом возможны следующие три варианта.

1. Если $t + \theta > \varphi'(a, t, m, p, r, \theta)$, то следующим моментом принятия решения будет $\varphi'(\alpha, t, m, p, r, \theta)$, свободным по состоянию на этот момент времени окажется по-прежнему процессор P^m; процессор $P^{(3-m)}$ по состоянию на очередной момент принятия решения будет продолжать обслуживать объект с номером r и до конца этого процесса останется $t + \theta - \varphi'(\alpha, t, m, p, r, \theta)$ тактов дискретного времени.

2. Если $t + q < \varphi'(\alpha, t, m, p, r, \theta)$, то очередным моментом принятия решения будет $t + \theta$, свободным по состоянию на этот момент времени окажется процессор $P^{(3-m)}$; процессор P^m будет продолжать выполнять обслуживание объекта, до завершения которого останется $\varphi'(\alpha, t, m, p, r, \theta) - (t + \theta)$ тактов дискретного времени.

3. Если $t + \theta = \varphi'(\alpha, t, m, p, r, \theta)$, то в следующий за t момент принятия решения ситуация будет описываться либо набором $(t + \theta, 1, \mu(a, t + \theta), \mu(r, t + \theta), 0)$, либо набором $(t + \theta, 1, \mu(r, t + \theta), \mu(a, t + \theta), 0)$ для случаев $m = 1$ и $m = 2$ соответственно.

Для разделения описанных вариантов введем в рассмотрение функции $x(\alpha, t, m, p, r, \theta)$, $y(\alpha, t, m, p, r, \theta)$, $z_1(a, t, m, p, r, \theta)$ и $z_2(\alpha, t, m, p, r, \theta)$, которые будут принимать значения 1 или 0 в перечисленных выше четырех случаях соответственно. Также введем в рассмотрение функцию Беллмана $D(t, m, p, r, \theta)$, каждое значение которой суть максимально возможная суммарная прибыль по всем обслуживаемым объектам, начиная с момента времени t при условии, что ситуация характеризуется набором (t, m, p, r, θ). Очевидно, что $D(0, 1, x_1, x_2, 0)$ – максимально возможный суммарный доход по всем объектам потока O.

С учетом вышеописанного на основании принципа оптимальности [3] запишем следующее рекуррентное соотношение динамического программирования:

$$D(t, m, p, r, \theta) = \max_{\alpha \in U(t,m,p,r,\theta)} \{w^m(\alpha) + x(\alpha, t, m, p, r, \theta) \cdot D(\varphi'(\alpha, t, m, p, r, \theta), m, \lambda(\alpha, t, m, p, r, \theta)), r, t+\theta - \varphi'(\alpha, t, m, p, r, \theta)) + y(\alpha, t, m, p, r, \theta) \cdot D(t + \theta, 1 - m, \gamma(r, t + \theta)), m, \varphi'(\alpha, t, m, p, r, \theta) - t + \theta) + z_1(\alpha, t, m, p, r, \theta) \cdot D(t + q, 1, \gamma(\alpha, t + \theta), \gamma(r, t + \theta), 0) + z_2(\alpha, t, m, p, r, \theta) \cdot D(t + \theta, 1, \gamma(r, t + \theta), \gamma(\alpha, t + \theta), 0)\}. \quad (8)$$

Пусть T – момент времени, когда последний объект потока O покидает зону обслуживания (по состоянию на момент времени T оба процессора должны быть свободны). Как очевидно, для любой целой неотрицательной константы C и любых целочисленных значений параметров p и r имеет место соотношение

$$D(T + C, 1, p, r, 0) = 0. \quad (9)$$

Формулы (8), (9) – рекуррентные соотношения динамического программирования для решения рассматриваемой экстремальной задачи.

Процесс вычислений целесообразно начинать с процедуры предварительной разметки, когда определяются наборы значений аргументов, для которых подсчет функции Беллмана действительно необходим. Разметка выполняется в порядке возрастания значения аргумента t вплоть до достижения наборов, когда его значение оказывается не меньшим, чем T. Затем по формулам (9), (10) последовательно в порядке убывания значений аргумента t определяем значения функции Беллмана для всех вычисленных в процессе разметки наборов. Последней вычисляется величина $D(0, 1, x_1, x_2, 0)$.

Оценка временнóй вычислительной сложности процедуры синтеза оптимальной стратегии обслуживания

Определим оценку вычислительной сложности построенного алгоритма. Последняя связана с количеством вычисляемых значений функции Беллмана $D(t, m, p, r, q)$ и трудоемкостью нахождения каждого очередного ее значения. Первый аргумент функции может принимать порядка T различных значений. Считаем, что величина T не превышает значений некоторой линейно зависящей от n функции; полагаем величину l и максимальную из продолжительностей обслуживания объектов процессорами заранее заданными константами. Легко видеть, что при вычислении каждого очередного значения функции $D(t, m, p, r, q)$ по формуле (9) количество выполняемых элементарных операций не превышает значений некоторой линейно зависящей от n функции. В таком случае алгоритм синтеза оптимальной стратегии r^* оценивается сверху величиной $O(n^3)$.

Как показали вычислительные эксперименты, предварительная разметка позволяет существенно уменьшить число рассчитываемых значений функции Беллмана, по сравнению с максимальным. Так, на ПК с типовыми характеристиками синтез оптимальной стратегии обслуживания для потока из 20 объектов происходил в среднем за 25 с. Вместе с тем для некоторых сочетаний параметров системы обслуживания синтез длился более 200 с. Объем используемой алгоритмом оперативной памяти оценивается сверху величиной $O(n^2)$ и не является ограничивающим фактором при решении прикладных задач рассматриваемого типа.

Заключение

В качестве непосредственных обобщений для транспортных приложений значимыми являются модели, учитывающие в различных сочетаниях (в зависимости от реально складывающейся обстановки) следующие модификации:

а) каждый объект потока O поступает в рабочую зону Ξ через «свою» внутреннюю точку (отрезка AB);
б) объект $o(i)$ может быть принят на обслуживание, если получаемая при этом прибыль не ниже заданного порогового значения d_i ($d_i > 0$, $i \in \{1, 2,\dots,n\}$);
в) в потоке O имеются объекты с привилегированным статусом, подлежащие обслуживанию в том числе и в случае отрицательной прибыли по каждому из них;

г) учитываются расходы на перемещения процессоров и решается задача синтеза стратегии обслуживания объектов потока, обеспечивающей максимум прибыли;

д) учитываются расходы на перемещения процессоров и решается задача синтеза оптимальной стратегии обслуживания объектов потока с двумя независимыми критериями [4], один из которых оценивает доход от обслуживания, а другой – расходы, связанные с перемещением процессоров;

е) синтез субоптимальных стратегий обслуживания осуществляется на основе концепции d-расписаний [5], позволяющей существенно повысить скорость работы решающего алгоритма;

ж) рабочая зона Ξ представляется связным неориентированным графом, часть вершин которого соответствует точкам входа и выхода из зоны обслуживания объектов мультипотока.

Список литературы

1. Коган Д.И., Федосенко Ю.С., Шеянов А.В. Проблема синтеза оптимального расписания обслуживания бинарного потока объектов mobile-процес-сором // Труды III Международной конференции «Дискретные модели в теории управляющих систем». – М.: Изд-во МГУ им. М.В. Ломоносова, 1998. – С. 43–46.

2. Коган Д.И., Федосенко Ю.С. Задача синтеза оптимального расписания обслуживания бинарного потока объектов в рабочей зоне mobile-процессора // Вестник Нижегородского университета. Математическое моделирование и оптимальное управление. – 1999. – Вып. 1(20). – С. 179–187.

3. Беллман Р., Дрейфус С. Динамическое

THE OPTIMAL STRATEGY OF SERVICING BINARY-OBJECT FLOW BY TWO MOBILE-PROCESSORS IN A LINEAR WORKING AREA

M.B. Reznikov, Yu.S. Fedosenko

We consider the problem of synthesizing the optimal strategy of single-phase servicing of a binary-object flow by two mobile-processors. The summary profit is used as the optimization criterion. The mathematical model of servicing is formulated, the algorithm for synthesizing the optimal servicing strategy is developed, and the computational complexity of this is estimated. The results of numerical experiments are presented.

программирование. – М.: ИЛ, 1960. – 400 с.

4. Коган Д.И. Динамическое программирование и дискретная многокритериальная оптимизация. – Н. Новгород: Изд-во Нижегородского ун-та, 2005. – 260 с.

5. Коган Д.И., Федосенко Ю.С. Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы // Дискретная математика. – 1996. – Т. 8, № 3. – С. 135–147.