Алгоритмы синтеза стратегий в трехкритериальной задаче однопроцессорного обслуживания группировки стационарных объектов

А.М. Пушкин[1], Д.И. Коган[1], Ю.С. Федосенко[2]

[1]Московский государственный университет информационных технологий, радиотехники и электроники
[2]Волжский государственный университет водного транспорта, Нижний Новгород

Аннотация: Рассматривается модель обслуживания группы стационарных объектов, расположенных вдоль одномерной рабочей зоны перемещающегося процессора. Процессор выполняет одностадийный цикл обслуживания без прерываний, который начинается и заканчивается в базовой точке. Для каждого объекта считаются заданными его местоположение, требуемая продолжительность обслуживания, ранний срок начала обслуживания и функция индивидуального штрафа. Изучается многокритериальная задача, где в качестве минимизируемых критериев выступают: общее пройденное процессором расстояние, момент возвращения процессора в базовую точку после обслуживания всех объектов и величина суммарного по всем объектам штрафа. Решение выполняется с использованием соотношений, основанных на принципе динамического программирования, а также на основе идеологии эволюционно-генетических вычислений. При использовании динамического программирования изучаются вопросы построения полных совокупностей эффективных оценок. Исследуются вопросы сравнения двух множеств оценок, приводятся оценки вычислительной сложности, пример реализации и результаты экспериментов.
Ключевые слова: многокритериальная оптимизация, стратегии обслуживания, динамическое программирование, эволюционные вычисления, NP-трудность, мобильный процессор, стационарные объекты.

Введение

Рассматриваемая в статье модель предназначена для описания функционирования локальных логистических систем [1,2], в которых мобильный процессор осуществляет обслуживание рассредоточенной в рабочей зоне группировки стационарных объектов.

В ранее опубликованных работах [3,4,5] по обслуживанию группировки объектов априорно учитываются обусловленные спецификой конкретных приложений те или иные ограничения на перемещения процессора в одномерной рабочей зоне. Так, в [3,4] рассматривается модель обслуживания при реализации процессором двух рейсов – от начальной до конечной точки зоны и обратно. Модель, изучаемая в [5], требует обслуживания всех объектов группировки при реализации процессором только прямого рейса − от начальной точки зоны к её конечной точке; при этом разрешены ограниченные по глубине возвратные движения процессора для обслуживания «пропущенных» объектов. Для синтеза эффективных стратегий обслуживания в указанных работах сформулированы бикритериальные [6] задачи оптимизации и в рамках парадигмы Парето [7] сконструированы решающие алгоритмы динамического программирования [8].

Ниже исследуемая трехкритериальная [9] задача оптимизации стратегий управления обслуживанием формулируется в разделе 1 в рамках обобщенной математической модели, свободной от каких-либо ограничений на перемещения процессора в его рабочей зоне. Такая модель существенно расширяет сферу покрываемых логистических приложений и создает теоретическую базу для разработки достаточно универсальной информационной системы поддержки управления обслуживанием группировки стационарных объектов, рассредоточенных в одномерной рабочей зоне процессора.

Конструируемый в разделе 2 решающий алгоритм DP основан на концепциях эффективных оценок [7], многокритериального динамического программирования [10]. Здесь же приведен численный пример реализации алгоритма и результаты сравнительных вычислительных исследований алгоритмов динамического программирования и сконструированного по типовой схеме алгоритма GA, основанного на эволюционно-генетической концепции [11].

В заключительном разделе статьи обсуждаются полученные результаты и возможные направления развития исследований рассматриваемого класса прикладных задач в интересах создания компьютерных систем поддержки логистических процессов.

Математическая модель и постановка задачи

Полагается заданной совокупность $O_n = \{o_1, o_2, \ldots, o_n\}$ подлежащих однофазному обслуживанию стационарных объектов, расположенных соответственно в точках $1, 2, \ldots, n$ одномерной рабочей зоны L обслуживающего процессора P. Зона L конечна, её начальная точка A является базовой для процессора, присвоим ей номер 0. Подлежащие обслуживанию объекты считаем пронумерованными в порядке возрастания их расстояний от точки A. Конечная точка B зоны L является местом расположения объекта o_n. Для параметров модели обслуживания объектов o_j, $j = \overline{1, n}$, примем следующие обозначения:

τ_j – продолжительность обслуживания процессором P объекта o_j;

r_j – ранний срок начала обслуживания (момент готовности к обслуживанию) объекта o_j;

$s_{j-1,j}$ – расстояние между точками $j-1$ и j;

$\gamma_{j-1,j}$ и $\gamma_{j,j-1}$ – продолжительности перемещения процессора P соответственно от точки $j-1$ к точке j и от точки j к точке $j-1$;

$\varphi_j(t)$ – монотонно возрастающая в нестрогом смысле функция индивидуального штрафа по объекту o_j от момента t завершения его обслуживания;

$\gamma_{p,q}$ – продолжительность перемещения процессора от точки p к точке q: $\gamma_{p,q} = \sum_{j=p}^{q-1} \gamma_{j,j+1}$, если $p < q$ и $\gamma_{p,q} = \sum_{j=q}^{p-1} \gamma_{j+1,j}$, если $p > q$;

$s_{p,q}$ – расстояние между точками p и q: $s_{p,q}=\sum_{j=p}^{q-1}s_{j,j+1}$, если $p<q$ и $s_{p,q}=\sum_{j=q}^{p-1}s_{j,j+1}$, если $p>q$.

Считаем параметры τ_j, $s_{j-1,j}$, $\gamma_{j-1,j}$, $\gamma_{j,j-1}$ принимающими натуральные значения, а параметр r_j принимающим только целые неотрицательные значения. Процессор P начинает движение из базовой точки A в момент времени $t=0$; по завершению обслуживания всех объектов группировки O_n он возвращается в точку A. Обслуживание каждого объекта осуществляется процессором P однократно, без прерываний; одновременное обслуживание процессором двух и более объектов запрещено.

Стратегией обслуживания будем называть произвольную упорядоченную по возрастанию последовательность индексов $S=(i_1,i_2,...,i_n)$ из совокупности $N=\{1,2,....,n\}$, определяющую порядок обслуживания объектов. Реализующими стратегию S расписаниями обслуживания ρ именуем кортежи вида $\rho=(i_1,a_{i_1},b_{i_1}),...,(i_k,a_{i_k},b_{i_k}),...,(i_n,a_{i_n},b_{i_n})$, в которых a_{i_k} и b_{i_k} – соответственно моменты начала и завершения обслуживания объекта o_{i_k}, $k=\overline{1,n}$.

Не связанные с обслуживанием объекта o_j и ожиданием момента наступления раннего срока r_j, $j=\overline{1,n}$ начала его обслуживания промежуточные простои процессора запрещены, т.е. ниже будут рассматриваться только стратегии S, которым соответствуют компактные [12] r-расписания. Совокупность таких стратегий будем обозначать символом R.

В зависимости от складывающейся оперативной обстановки эффективность той или иной стратегии обслуживания в рассматриваемом классе локальных логистических систем может оцениваться ЛПР с учетом различных обстоятельств, среди которых основными являются следующие: $K_1(S)$ – суммарное пройденное процессором расстояние до момента возврата процессора в точку A; $K_2(S)$ – суммарное время работы процессора; $K_3(S)$ – суммарный по всем объектам штраф за простои в ожидании обслуживания. Соответствующая трехкритериальная задача имеет вид

$$\min_{S\in R}\{K_1(S), K_2(S), K_3(S)\}. \qquad (1)$$

При решении задачи (1) принимается концепция Парето, предусматривающая отыскание полной совокупности эффективных в ней оценок [7]. Соответствующие алгоритмы конструируются на основе концепции эффективных оценок [7].

Синтез стратегий обслуживания

Введем следующие обозначения: Z – общая задача (1), $Z(V,i)$ – частная задача, в которой вышедший из начальной точки A процессор сначала обслуживает совокупность объектов с индексами из произвольного подмножества $V\subset N$, а затем, в заключение обслуживает объект o_i, $i\notin V$.

Пусть $E(V,i)$– совокупность эффективных оценок в задаче $Z(V,i)$. Если V – пустое множество, то

$$E(\{\varnothing\},i)=(s_{0,i}, t_i, \varphi_i(t_i)),\ t_i=\max(\gamma_{0,i},r_i)+\tau_i,\ i\in N. \qquad (2)$$

Введем функцию $\Psi_{j,i}(M)$, которая преобразует каждый из трехмерных векторов множества M по правилу $(a,b,c)\to(a+s_{j,i}, t_{j,i}, c+\varphi_i(t_{j,i}))$, где $t_{j,i}=\max(b+\gamma_{j,i},r_i)+\tau_i$. Считается, что данное правило работает в следующем предположении: конструируемая в процессе работы алгоритма стратегия обслуживания на конкретной итерации завершается в точке j

рабочей зоны и имеет оценку (a,b,c), после чего из этой точки совершается переход в точку i зоны с пересчетом оценки согласно указанному правилу.

Предположим, что множества оценок $E(V,i)$ для всех $i \in N$ и всех возможных k-элементных множеств V уже вычислены (k – фиксированная константа, $k < n$). Тогда значение $E(V',i)$, где V' – любое $(k+1)$-элементное подмножество совокупности $N \setminus \{i\}$, вычисляется по формулам

$$E(V',i) = eff\left\{\bigcup_{j\in V'}\left[\Psi_{j,i}\left(E\left(V' \setminus \{j\}, j\right)\right)\right]\right\}. \tag{3}$$

Пусть все объекты прошли обслуживание. Тогда процессору необходимо совершить переход от последнего обслуженного объекта $\alpha, \alpha \in N$ в базовую точку A. В этом случае полная совокупность эффективных оценок E_{full} в задаче Z находится из соотношения

$$E_{full} = eff\left\{\bigcup_{\alpha\in N}\left[E(N \setminus \alpha, \alpha) \oplus \left(s_{\alpha,0}, \gamma_{\alpha,0}, 0\right)\right]\right\}, \tag{4}$$

в котором операция $\oplus$ определяется следующим образом:

$$\{(a_1,b_1,c_1),\ldots,(a_l,b_l,c_l)\} \oplus (d,e,f) =$$
$$= \{(a_1+d, b_1+e, c_1+f),\ldots,(a_l+d, b_l+e, c_l+f)\}.$$

В процессе решения на каждой итерации отмечается индекс вершины перехода процессора и сохраняется полный путь, который он прошел для достижения данного набора эффективных оценок.

Формулы (2)-(4) – рекуррентные соотношения многокритериального динамического программирования [8,10] для решения задачи (1). Реализующие эти соотношения алгоритм DP имеет экспоненциальную оценку вычислительной сложности.

Пример выполнения алгоритма DP. Требуется найти полную совокупность эффективных оценок для задачи (1) с исходными данными, представленными в таблице 1.

Таблица № 1

Исходные данные задачи

l	$s_{l-1,l}$	$\gamma_{l-1,1}$	$\gamma_{l,l-1}$	τ_l	r_l	$\varphi_l(t)$
1	4	4	1	3	17	$6t$
2	4	2	2	10	11	$3t$
3	2	2	1	4	10	$5t$
4	3	1	1	10	7	t

Процедуру решения представляем как процесс заполнения таблицы 2 и последующих, каждая следующая таблица соответствует очередной выполненной итерации (под итерацией понимается набор вычислений множеств $E(V,i)$ при условии, что $|V|=k$).

В процессе вычислений по рекуррентным соотношениям (2) и (3) значение k последовательно принимает значения $0,1,\ldots,n-1$. Последняя, выполняемая по формуле (4), итерация синтезирует полную совокупность эффективных оценок в задаче (1).

В таблицах рядом с каждой эффективной оценкой в той же строке указывается порождающая ее стратегия S^*. Стратегия S^* отличается от других допустимых стратегий обслуживания тем, что после обслуживания перечисленных в ней объектов процессор не совершает переход в базовый пункт A.

Согласно формуле (2) в таблице 2 запишем результаты первой итерации.

Таблица № 2

Результаты итерации 1

V	i	$E(V,i)$	S^*
{Ø}	1	(4, 20, 120)	(1)
{Ø}	2	(8, 21, 63)	(2)
{Ø}	3	(10, 14, 70)	(3)
{Ø}	4	(13, 19, 19)	(4)

Для иллюстрации ниже приведем цепочку вычислений для нахождения совокупности $E(\{\varnothing\},2)$: $t_2 = \max(\gamma_{0,2}, r_2) + \tau_2 = \max(6,11) + 10 = 21$, $E(\{\varnothing\},2) = (s_{0,2}, t_2, \varphi_2(t_2)) = (8,21,63)$.

Применяя формулу (4), в таблице 3 запишем результаты второй итерации.

Таблица № 3

Результаты итерации 2

V	i	$E(V,i)$	S^*
{1}	2	(8, 32, 216)	(1, 2)
{1}	3	(10, 28, 260)	(1, 3)
{1}	4	(13, 35, 155)	(1, 4)
{2}	1	(12, 26, 219)	(2, 1)
{2}	3	(10, 27, 198)	(2, 3)
{2}	4	(13, 34, 97)	(2, 4)
{3}	1	(16, 20, 190)	(3, 1)
{3}	2	(12, 25, 145)	(3, 2)
{3}	4	(13, 25, 95)	(3, 4)
{4}	1	(22, 26, 175)	(4, 1)
{4}	2	(18, 31, 112)	(4, 2)
{4}	3	(16, 24, 139)	(4, 3)

Иллюстрирующая цепочка вычислений для нахождения совокупности $E(\{2\},4)$ выглядит следующим образом:

$$eff\left\{\Psi_{2,4}\left(\left(8,21,63\right)\right)\right\} = eff\left\{\left(8,21,63\right) \to \left(8+s_{2,4},t_{2,4},63+\varphi_4\left(t_{2,4}\right)\right)\right\},$$

$$t_{2,4} = \max\left(21+\gamma_{2,4},r_4\right)+\tau_4 = \max\left(21+3,7\right)+10 = 34,$$

$$eff\left\{\Psi_{23}\left(\left(12,72\right)\right)\right\} = eff\left\{\left(8+5,34,63+34\cdot 1\right)\right\} = \left(13,34,97\right).$$

Аналогично второй итерации в таблице 4 представлены результаты третьей итерации.

Таблица № 4

Результаты итерации 3

V	i	$E(V,i)$	S^*
{1, 2}	3	(10, 38, 406) (18, 34, 389)	(1, 2, 3) (2, 1, 3)
{1, 2}	4	(21, 41, 260) (13, 45, 261)	(2, 1, 4) (1, 2, 4)
{1, 3}	2	(20, 32, 286) (12, 39, 377)	(3, 1, 2) (1, 3, 2)
{1, 3}	4	(25, 35, 225) (13, 39, 299)	(3, 1, 4) (1, 3, 4)
{1, 4}	2	(18, 47, 296) (26, 38, 289)	(1, 4, 2) (4, 1, 2)
{1, 4}	3	(28, 34, 345) (16, 40, 355)	(4, 1, 3) (1, 4, 3)
{2, 3}	1	(16, 30, 325)	(3, 2, 1)
{2, 3}	4	(13, 38, 236) (17, 38, 183)	(2, 3, 4) (3, 2, 4)
{2, 4}	1	(22, 36, 328)	(4, 2, 1)
{2, 4}	3	(16, 39, 292) (20, 37, 297)	(2, 4, 3) (4, 2, 3)
{3, 4}	1	(22, 32, 287) (22, 30, 319)	(3, 4, 1) (4, 3, 1)
{3, 4}	2	(18, 35, 244) (18, 37, 206)	(4, 3, 2) (3, 4, 2)

Результаты четвертой итерации приведены в таблице 5.

Таблица № 5

Результаты итерации 4

V	i	$E(V,i)$	S^*
{1, 2, 3}	4	(13, 49, 455) (17, 52, 429) (25, 45, 331) (21, 45, 434)	(1, 2, 3, 4) (1, 3, 2, 4) (3, 1, 2, 4) (2, 1, 3, 4)
{1, 2, 4}	3	(28, 44, 509) (24, 46, 490) (16, 50, 511)	(4, 1, 2, 3) (2, 1, 4, 3) (1, 2, 4, 3)
{1, 3, 4}	2	(30, 47, 366) (26, 44, 419) (18, 51, 452) (26, 42, 445)	(3, 1, 4, 2) (3, 4, 1, 2) (1, 3, 4, 2) (4, 3, 1, 2)
{2, 3, 4}	1	(22, 40, 484) (26, 45, 453) (22, 42, 458)	(4, 3, 2, 1) (3, 2, 4, 1) (3, 4, 2, 1)

Последняя итерация, реализуемая по формуле (5), отображает факт перехода процессора в базовую точку после завершения обслуживания всех объектов. Финальные оценки работы процессора и соответствующие им стратегии обслуживания представлены в таблице 6.

Таблица № 6

Результаты итерации 5

E_{full}	S
(26, 41, 484)	(4, 3, 2, 1)
(26, 43, 458)	(3, 4, 2, 1)
(26, 54, 452)	(1, 3, 4, 2)
(30, 57, 429)	(1, 3, 2, 4)
(30, 46, 453)	(3, 2, 4, 1)
(34, 45, 445)	(4, 3, 1, 2)
(34, 47, 419)	(3, 4, 1, 2)
(38, 50, 331)	(3, 1, 2, 4)

В итоге построена полная совокупность эффективных оценок, и для каждой такой оценки определено соответствующее расписание обслуживания. Из построенной совокупности эффективных оценок ЛПР осуществляет свой выбор, принимая во внимание обстоятельства, не предусмотренные построенной в разделе 1 математической моделью.

Отметим, что задача (1) относится к числу труднорешаемых [13] в силу того, что указанным свойством обладают некоторые её частные конкретизации. В качестве одной из таковых отметим бикритериальную задачу

$$\min_{S\in R}\left\{K_1(S), K_3(S)\right\},$$

которая NP-трудна [13] даже в условии принятия для неё схемы лексикографического упорядочения критериев с ведущим критерием $K_1(S)$ и линейными функциями индивидуального штрафа. Указанный результат установлен в [4].

При создании компьютерных средств поддержки оперативного управления обслуживанием потоков объектов в локальных логистических системах существенным обстоятельством, которое приходится учитывать при разработке математического обеспечения таких средств, являются жесткие ограничения [3,4,12] на допускаемую штатным регламентом длительность формирования стратегий обслуживания; обычно этот промежуток времени не должен превышать 15 минут. В силу этого задача (1) для ряда практических значимых размерностей потока O_n оказывается нерешаемой, и, следовательно, актуальной является проблема построения алгоритмов, позволяющих генерировать субоптимальные стратегии обслуживания за приемлемое для приложений время.

В таблице 7 приведены оценки решающего задачу (1) алгоритма GA, реализующего типовую вычислительную схему эволюционно-генетической концепции [11]. В качестве оценок были взяты: T_{mean} – средняя продолжительность решения задачи (1); K_{eff} – минимальное, максимальное и среднее количество эффективных («субэффективных» для GA) оценок.

Выбираемые для вычислений значения параметров модели соответствовали равномерному закону распределения на следующих обусловленных физическим смыслом диапазонах замкнутых интервалов: $s_{i-1,i} \in [1,5]$, $\gamma_{i-1,i} \in [5,20]$, $\gamma_{i,i-1} \in [5,20]$, $\tau_i \in [1,10]$, $r_i \in [10, 10 \cdot n]$, $a_i \in [1,10]$, где n – размерность задачи, a_i – коэффициент линейной функции штрафа $\varphi_i(t) = a_i \cdot t$.

Для сравнения в этой же таблице приведены значения показателей T_{mean} и K_{eff}, полученные при реализации алгоритма DP.

Таблица № 7

Результаты вычислительных экспериментов

n	T_{mean}, с		K_{eff}		$\Delta,\%$
	GA	DP	GA	DP	
5	0	0	1 – 5 (2)	1 – 5 (2)	0
10	16	2	1 – 8 (3)	1 – 9 (3)	3
15	180	90	1 – 6 (3)	2 – 9 (4)	10
17	360	660	1 – 8 (4)	1 – 12 (5)	20
19	660	5400	1 – 5 (2)	2 – 10 (4)	26
20	900	15300	1 – 7 (3)	3 – 11 (5)	36

Приведенный в таблице 7 показатель Δ используется для интегральной оценки отклонения множества «субэффективных» оценок от точного множества оценок. При определении значений этого показателя исходим из следующих соображений.

Пусть W – множество эффективных оценок, полученных при решении задачи (1) алгоритмом DP, Q – множество оценок, полученных при решении задачи алгоритмом GA, в общем случае $|W| \neq |Q|$; w_i, $i = \overline{1,|W|}$, q_j, $j = \overline{1,|Q|}$ –

элементы соответствующих множеств. Значения критериев $K_1(S)$, $K_2(S)$, $K_3(S)$ при фиксированном S можно рассматривать, как декартовы координаты точки в трехмерном пространстве, при этом точка $O(0,0,0)$ соответствует началу координат. Через $|C_1C_2|$ обозначаем расстояние между точками (оценками) C_1 и C_2; отклонение Δ вычисляем по формуле

$$\Delta = \max_i \left(\min_j \left(\frac{|w_i q_j|}{|w_i O|} \right) \right) \cdot 100,\ \%.$$

Программная реализация алгоритмов DP и GA была выполнена на высокоуровневом языке программирования общего назначения Python 2.7, вычислительные эксперименты проводились на ПК с процессором Intel® Core™ i7 4.20 GHz и оперативной памятью объемом 16 Gb.

Как следует из таблицы 7, для задачи (1) с размерностью потока n не более 17, следует использовать алгоритм DP; в случае больших значений n целесообразно использовать алгоритм GA.

Результаты

Авторами сформулирована и изучена трехкритериальная задача обслуживания мобильным процессором линейно рассредоточенной группировки стационарных объектов с учетом предписанных ранних сроков их готовности к обслуживанию. При решении принята концепция Парето; совокупность эффективных в задаче оценок строится с использованием рекуррентных соотношений многокритериального динамического программирования. Полученный теоретический результат о вычислительной сложности соответствует оценке трудоёмкости решающего задачу алгоритма.

Выполненные вычислительные эксперименты и сравнение характеристик разработанного алгоритма динамического программирования с аналогичными характеристиками эволюционно-генетического алгоритма приближенного синтеза стратегий обслуживания позволили установить области их целесообразного применения.

В направлении развития рассматриваемого класса прикладных задач представляется целесообразной разработка и экспериментальное исследование алгоритмов синтеза стратегий обслуживания в локальных логистических системах с существенно более высокими характеристиками отработки, в том числе основанных на иных метаэвристических парадигмах.

Статья подготовлена по результатам исследований, выполненных при финансовой поддержке Российского фонда фундаментальных исследований в рамках проекта № 15-07-03141.

Литература

1. Кочерга В.Г., Зырянов В.В., Хачатурян А.В. Планирование и организация грузовых автомобильных перевозок на улично-дорожной сети мегаполисов // Инженерный вестник Дона, 2012, №2, URL: ivdon.ru/ru/magazine/archive/n2y2012/869.

2. Шегельман И.Р., Кузнецов А.В., Скрыпник В.И., Баклагин В.Н. Методика оптимизаций транспортно-технологического освоения лесосырьевой базы с минимизацией затрат на заготовку и вывозку древесины // Инженерный вестник Дона, 2012, №4 (ч.2) URL: ivdon.ru/ru/magazine/archive/n4p2y2012/1284.

3. Коган Д.И., Федосенко Ю.С., Дуничкина Н.А. Бикритериальные задачи обслуживания стационарных объектов в одномерной рабочей зоне процессора // Автоматика и телемеханика, 2012, №10, с. 93-110.

4. Дуничкина Н.А., Коган Д И., Пушкин А.М., Федосенко Ю.С. Об одной модели обслуживания стационарных объектов перемещающимся в одномерной рабочей зоне процессором // Труды XII Всероссийского совещания по проблемам управления. М.: ИПУ РАН, 2014, с. 5044-5052.

5. Пушкин А.М. Модель обслуживания стационарных объектов перемещающимся процессором с возможностью возвратов // Научно-технический вестник Поволжья, 2014, №5, с. 288-292.

6. Steuer, R. Multiple Criteria Optimization: Theory, Computation and Application, Russian Edition, Radio e Svyaz, Moscow, 1992, 504 p.

7. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач. М.: Наука, 1982. 256 с.

8. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965. 460 с.

9. T'kindt V., Billaut J. Multicriteria scheduling: models and algorithms. Springer, 2006. 359 p.

10. Коган Д.И. Динамическое программирование и дискретная многокритериальная оптимизация. Нижний Новгород: Изд-во ННГУ, 2005. 260 с.

11. Батищев Д.И., Костюков В.Е., Неймарк Е.А., Старостин Н.В. Решение дискретных задач с помощью эволюционно-генетических алгоритмов. Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2011. 199 с.

12. Коган Д.И., Федосенко Ю.С. Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы // Дискретная математика, 1996, т. 8, вып. 3, с. 135-147.

13. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.

References

1. Kocherga V.G., Zyrjanov V.V., Hachaturjan A.V. Inženernyj vestnik Dona (Rus), 2012, №2, URL: ivdon.ru/ru/magazine/archive/n2y2012/869.

2. Shegel'man I.R., Kuznecov A.V., Skrypnik V.I., Baklagin V.N. Inženernyj vestnik Dona (Rus), 2012, №4 (p.2) URL: ivdon.ru/ru/magazine/archive/n4p2y2012/1284.

3. Kogan D.I., Fedosenko Yu.S., Dunichkina N.A. Avtomatika i telemekhanika, 2012, №10, pp. 93-110.

4. Dunichkina N.A., Kogan D.I., Pushkin A.M., Fedosenko Yu.S. Trudy XII Vserossiyskogo soveshchaniya po problemam upravleniya. M.: IPU RAN, 2014, pp. 5044-5052.

5. Pushkin A.M. Nauchno-tekhnicheskiy vestnik Povolzh'ya, 2014, №5, pp. 288-292.

6. Steuer, R. Multiple Criteria Optimization: Theory, Computation and Application, Russian Edition, Radio e Svyaz, Moscow, 1992, 504 p.

7. Podinovskiy V.V., Nogin V.D. Pareto-optimal'nye resheniya mnogokriterial'nykh zadach [Pareto-optimal solutions of multiobjective problems]. M.: Nauka, 1982. 256 p.

8. Bellman R., Dreyfus S. Prikladnye zadachi dinamicheskogo programmirovaniya [Applications of dynamic programming]. M.: Nauka, 1965. 460 p.

9. T'kindt V., Billaut J. Multicriteria scheduling: models and algorithms. Springer, 2006. 359 p.

10. Kogan D.I. Dinamicheskoe programmirovanie i diskretnaya mnogokriterial'naya optimizatsiya [Dynamic programming and discrete multi-objective optimization]. Nizhniy Novgorod: Izd-vo NNGU, 2005. 260 p.

11. Batishchev D.I., Kostyukov V.E., Neymark E.A., Starostin N.V. Reshenie diskretnykh zadach s pomoshch'yu evolyutsionno-geneticheskikh algoritmov [Solution of discrete tasks using genetic algorithms]. Nizhniy Novgorod: Izd-vo NNGU im. N.I. Lobachevskogo, 2011. 199 p.

12. Kogan D.I., Fedosenko Yu.S. Diskretnaya matematika, 1996, t. 8, vyp. 3, pp. 135-147.

13. Geri M., Dzhonson D. Vychislitel'nye mashiny i trudnoreshaemye zadachi [Computers and intractable problems]. M.: Mir, 1982. 416 p.