

УДК:519.854.3

**ЗАДАЧИ И АЛГОРИТМЫ ОПТИМИЗАЦИИ РАСПИСАНИЙ  
ПАРАЛЛЕЛЬНО-ПОСЛЕДОВАТЕЛЬНЫХ СИСТЕМ  
С НЕОПРЕДЕЛЕННЫМИ МАРШРУТАМИ ОБСЛУЖИВАНИЯ****Ю.А. Мезенцев, И.В. Эстрайх***Новосибирский государственный технический университет*

Представлен новый подход к решению проблем составления оптимальных по быстродействию расписаний параллельно-последовательных систем с использованием двухэтапной схемы: кластеризации и последующего определения маршрутов. Частными случаями применения разработанного инструментария являются решения различных модификаций задач маршрутизации с минимаксным критерием. Приведены формальные постановки рассматриваемых подзадач кластеризации и составления маршрутов в виде NP-трудных задач дискретного программирования. Предложен приближенный алгоритм решения поставленной задачи, основанный на ее декомпозиции на подзадачу оптимального по равномерной нагрузке на приборы разбиения множества заявок на подмножества и ряд подзадач определения последовательностей их обслуживания приборами. Приведены иллюстративные примеры применения развиваемого подхода, вычислены эмпирические оценки точности и быстродействия его программной реализации. На сгенерированных тестовых примерах в широком диапазоне размерностей исследованы быстродействие и точность реализованных алгоритмов. Обозначены возможные практические применения при построении календарных графиков реализации крупных промышленных проектов и определены направления развития предложенного подхода.

*Ключевые слова:* параллельно-последовательная система, неопределенные маршруты обслуживания, оптимальность по быстродействию, маршрутизация, кластеризация, задача коммивояжера.

DOI: 10.17212/1727-2769-2016-3-83-97

**Введение**

Авторы вынуждены несколько расширить классификацию задач теории расписаний (ЗТР), поскольку рассматриваемые в статье задачи не вписываются в существующие классификации, по смыслу явно относясь к ЗТР. Например, задача коммивояжера (ЗК) является частным случаем open shop problem (OSP) для одного прибора. Задачи же на составление маршрутов и оптимальных по быстродействию расписаний движения по этим маршрутам, являющиеся расширениями ЗК, в состав ЗТР обычно не включаются. Поэтому рассматриваемую ниже задачу оптимизации расписаний параллельно-последовательной системы с неопределенными маршрутами обслуживания для краткости обозначим как URPM (Undefined routes parallel machines problem). Данная задача с произвольным числом заявок и приборов в различных модификациях имеет множество теоретических обобщений и практических применений. Частные ее случаи (ЗК, задачи  $m$ -коммивояжеров, маршрутизации и связанные темы) с разным успехом разрабатываются более пяти десятилетий, не теряя актуальности в настоящее время [1–33]. Однако приемлемых по быстродействию средств точного решения общей, а также большинства порождаемых прикладных задач для реальных размерностей найти не удается в

---

Исследование выполнено при финансовой поддержке НГТУ (темплан НИР НГТУ), проект ТП-ЭИ-1\_15

силу их принадлежности к множеству труднорешаемых задач дискретной оптимизации (ДО). Разработанные же к настоящему времени (да и в перспективе тоже) вычислительные методы, как и большинство алгоритмов приближенного решения NP-полных задач ДО, обычно не имеют неулучшаемых оценок, и всегда впоследствии находится подобный алгоритм с лучшими оценками точности и (или) быстродействия, независимо от того, о каких оценках, априорных или апостериорных, идет речь. Этим, а также наличием многочисленных практических приложений обосновывается актуальность темы.

Настоящая статья посвящена уточнению названной URPM-проблемы с сужением и конкретизацией общей постановки относительно используемого критерия эффективности, расширением постановки относительно используемых ограничений, а также развитию соответствующих методов решения. В качестве критерия рассматривается общее быстродействие системы, что в изданных к настоящему времени работах встречается нечасто, возможно потому, что приводит к необходимости применения минимаксных целевых функций [28–33] с порождением задач трудоемкостей кратно превосходящих экспоненциальную относительно задачи той же размерности с линейным критерием. Кроме того требуют переосмысления и способы формализации, и порождаемые алгоритмы решения задачи. Ниже в подробностях эти постулаты раскрыты, приведены иллюстративные примеры и статистика результатов расчетов тестов для анализа точности и быстродействия реализующих алгоритмов.

### 1. Формальная постановка задачи

Рассматриваются система параллельных приборов разной производительности и множество заявок с заданными характеристиками (пунктами) обслуживания, оценками переходов (переналадки) для каждого из приборов (временем переналадки, ценами, штрафами или расстояниями) между всеми заявками (пунктами). Маршруты переходов (последовательности обслуживания) не заданы и подлежат определению. Обслуживание заявок приборами монопольное. Прерывания обслуживания приборами одних заявок в пользу других запрещены, запрещены и разрывы связности маршрутов.

URPM-проблема определения множества оптимальных по быстродействию расписаний обслуживания всей системы включает: назначение каждому из приборов некоторого подмножества заявок, непересекающегося с подмножествами заявок других приборов; определение порядков обслуживания (маршрутов переходов) для каждого из приборов всего множества заявок. Общая постановка сформулированной проблемы в терминах задачи дискретного программирования приводит к фатальному росту числа переменных и ограничений в сравнении с классической задачей коммивояжера [1–5].

Поэтому целесообразно рассматривать ее естественную декомпозицию как две подзадачи, по размерности сопоставимые с ЗК. Заметим, что подобный подход нов, и в литературе описаны два существенно различающихся способа его реализации. Во-первых, это построение общего связного маршрута (последовательности обслуживания) с последующим его разрезанием на  $m$  (по числу приборов) связных частей, см. например, [28]. Во-вторых, это разделение заявок (пунктов) на группы (алгоритмом дихотомического разделения, сортировкой по полярному углу и др.) и решение ЗК внутри каждой такой группы [33].

Для рассматриваемой задачи первый вариант неприменим вовсе, поскольку производительность разных приборов при обслуживании одних и тех же заявок в общем случае различна и построение начального связного маршрута по трудоем-

кости совпадает с решением общей URPM-проблемы. По этой причине за основу был принят второй способ решения, что, однако, потребовало существенной ревизии формальной постановки задачи. Ниже приведено описание подхода, в котором использованы наработки, полученные при решении задач оптимизации расписаний одностадийных параллельных и параллельно-последовательных систем с задержками начала обслуживания [34, 35].

Далее рассматриваются связанные подзадачи.

1. Кластеризация (наилучшее по быстродействию распределение заявок по  $m$  приборам). Фактически это задача о назначениях исходного множества приборов на множество заявок. Для каждой такой заявки и каждого прибора определены характеристики переходов от любых текущих к последующим заявкам (цена, время или расстояние). Маршруты переходов не фиксированы и подлежат определению. Пересечений приборов по маршрутам (и маршрутов по приборам) не допускается. Учитываются все прочие атрибуты заявленной URPM-проблемы, за исключением запрета на разрывы связности маршрутов.

2. Множество  $m$  ЗК ( $m$  подзадач коммивояжера). Быстродействие системы определяется наихудшей оценкой быстродействия из всех  $m$  ЗК.

Приведем формальную постановку подзадачи кластеризации.

Пусть все приборы и заявки пронумерованы:  $k = \overline{1, m}$ ,  $k$  – номер прибора,  $i, j = \overline{1, n}$  – номера заявок. Кроме этого для удобства обозначим через  $J$  множество заявок  $n$ ,  $J^k$  – подмножество заявок, назначаемых прибору  $k$ , а через  $n^k$  – число элементов каждого такого подмножества,  $J = \bigcup_{k=1}^m J^k$ ,  $\bigcap_{k=1}^m J^k = \emptyset$ . Также

введем следующие обозначения:  $s = \overline{1, S}$  – индекс используемого ресурса (в задачах маршрутизации),  $c_{i,j}^k$  – оценка (цена, время или расстояние) перехода (переналадки, перемещения) прибора  $k$  от обслуживания заявки  $i$  к обслуживанию заявки  $j$ . Первый из нижних индексов всегда обозначает заявку-источник, второй – заявку-сток. Будем полагать, что оценка (цена или время) процесса обслуживания заявки  $j$  (в классической ЗК равная нулю) уже учтена в  $c_{i,j}^k$ .

$$x_{i,j}^k = \begin{cases} 1, & \text{если имеет место переход прибора } k \\ & \text{от обслуживания заявки } i \text{ к } j, \\ 0, & \text{в противном случае, } i, j = \overline{1, n}, i \neq j, \end{cases}$$

$R_s^k$  – объем ресурса  $s$ , доступного прибору  $k$ . Например, в качестве ресурса может рассматриваться грузоподъемность автомобиля, тогда  $R_s^k$  – величина грузоподъемности.  $r_{j,s}^k$  – затраты ресурса  $s$  прибором  $k$  при обслуживании заявки  $j$ .

Требуется найти наилучшие разбиения  $\tilde{J}^k$ ,  $k = \overline{1, m}$ , и наилучшие назначения  $\tilde{x}_{i,j}^k$ ,  $i, j \in \tilde{J}^k$ , в терминах представленного ниже критерия быстродействия при выполнении приводимых далее условий. Выведем ограничения, обеспечивающие непересечение приборов по заявкам (маршрутам) и заявок по приборам:

$$\sum_{j=1}^n x_{i,j}^k = b_i^k, \quad k = \overline{1, m}, i = \overline{1, n}, \quad (1)$$

$$\sum_{i=1}^n x_{i,j}^k = d_j^k, \quad k = \overline{1,m}, j = \overline{1,n}, \quad (2)$$

$$\sum_{k=1,m}^{k \neq l} b_i^k + d_i^l \leq 1, \quad i = \overline{1,n}, \quad l = \overline{1,m}, \quad (3)$$

$$b_j^l + \sum_{k=1,m}^{k \neq l} d_j^k \leq 1, \quad j = \overline{1,n}, \quad l = \overline{1,m}. \quad (4)$$

В совокупности (1)–(4) означают следующее. Сумма числа назначений переходов от заявок-источников для всех приборов, кроме прибора  $l$ , к заявкам-стокам прибора  $l$  меньше либо равна единице. Это запрещает передачу заявок от всех приборов, кроме  $l$ , прибору  $l$ . Сумма числа назначений переходов от заявок-источников для прибора  $l$  к заявкам-стокам всех приборов, кроме прибора  $l$ , меньше либо равна единице. Это запрещает передачу заявок, назначенных прибору  $l$  любым другим приборам. Совместное выполнение всех этих условий гарантирует непересечение приборов по заявкам (и маршрутам) и заявок по приборам. Совмещение ограничений (1)–(4) с исключением зависимых переменных приводит к неравенствам (5) и (6):

$$\sum_{k=1,m}^{k \neq l} \sum_{j=1}^n x_{i,j}^k + \sum_{j=1}^n x_{j,i}^l \leq 1, \quad i = \overline{1,n}, \quad l = \overline{1,m}, \quad (5)$$

$$\sum_{i=1}^n x_{j,i}^l + \sum_{k=1,m}^{k \neq l} \sum_{i=1}^n x_{i,j}^k \leq 1, \quad j = \overline{1,n}, \quad l = \overline{1,m}. \quad (6)$$

Нижеследующие ограничения-равенства являются обычными условиями задачи о назначениях.

$$\sum_{k=1}^m \sum_{i=1}^n x_{i,j}^k = 1, \quad j = \overline{1,n}, \quad i \neq j, \quad (7)$$

$$\sum_{k=1}^m \sum_{j=1}^n x_{i,j}^k = 1, \quad j = \overline{1,n}, \quad i \neq j. \quad (8)$$

Дополняем задачу ограничениями на переменные (все переменные булевы):

$$x_{i,j}^k = \begin{cases} 1, & \text{если имеет место переход прибора } k \\ & \text{от обслуживания заявки } i \text{ к } j, \\ 0, & \text{в противном случае, } i, j = \overline{1,n}, \quad i \neq j. \end{cases} \quad (9)$$

И критерием быстродействия системы:

$$\sum_{j=1}^n \sum_{i=1}^n c_{i,j}^k x_{i,j}^k \leq \lambda, \quad k = \overline{1,m}, \quad (10)$$

$$\lambda \rightarrow \min. \quad (11)$$

В совокупности (10) и (11) реализуют минимаксный критерий, являющийся критерием быстродействия рассматриваемой параллельно-последовательной обслуживающей системы, или критерием равномерной нагрузки [34–36].

Кроме этого задачи маршрутизации, являющиеся расширениями одного из вариантов рассматриваемой URPM-проблемы, обычно дополняются ресурсными ограничениями вида

$$\sum_{j=1}^n \sum_{i=1}^n r_{j,s}^k x_{i,j}^k \leq R_s^k, \quad k = \overline{1, m}, \quad s = \overline{1, S}. \quad (12)$$

Оптимальные решения задач (5)–(11) и (5)–(12)  $\hat{J}^k$ ,  $\hat{x}_{i,j}^k$ ,  $i, j \in \hat{J}^k$ ,  $k = \overline{1, m}$ , определяют наилучшие из возможных нижние оценки  $\lambda_{\min}(\hat{x}_{i,j}^k)$  расписания всей URPM-проблемы. Для уточнения оценок и получения гарантированно связанных маршрутов обслуживания заявок необходимо решить  $m$  подзадач ЗК, для всех приборов  $k$ ,  $k = \overline{1, m}$ . Любая такая подзадача представима в виде:

$$\sum_{i \in \hat{J}^k} x_{i,j}^k = 1, \quad j \in \hat{J}^k, \quad i \neq j, \quad (13)$$

$$\sum_{j \in \hat{J}^k} x_{i,j}^k = 1, \quad i \in \hat{J}^k, \quad i \neq j, \quad (14)$$

$$u_i^k - u_j^k + n^k x_{i,j}^k \leq n^k - 1, \quad i, j = \overline{2, n^k}, \quad i \neq j, \quad (15)$$

$$u_i^k \geq 0, \quad i \in \hat{J}^k, \quad (16)$$

$$\Lambda^k = \sum_{j \in \hat{J}^k} \sum_{i \in \hat{J}^k} c_{i,j}^k x_{i,j}^k \rightarrow \min. \quad (17)$$

Выражение (15) предполагает для удобства, что индексы всех подмножеств заявок  $\hat{J}^k$ ,  $k = \overline{1, m}$ , перенумерованы от 1 до  $n^k$ . Набор оптимальных решений  $m$  подзадач ЗК (13)–(17)  $\hat{x}_{i,j}^k$ ,  $i, j \in \hat{J}^k$ ,  $k = \overline{1, m}$ , с оценкой  $\tilde{\Lambda} = \max \Lambda^k$  является приближенным решением рассматриваемой URPM-проблемы.

## 2. Процедуры поиска решений

Заметим, что обе представленные подзадачи принадлежат к одному классу NP-полных задач ДО. Этот тезис не требует доказательств, поскольку принадлежность к данному классу непосредственно вытекает из формальных постановок (5)–(12) и (13)–(17). Подзадача (13)–(17) является формализацией ЗК. Подзадача (5)–(12) содержит булевы переменные, «ранцевые» ограничения (10) и (12), а также логические ограничения (5) и (6), подматрица коэффициентов которых не образует фрагментов абсолютно унимодулярной матрицы. Эти свойства с избытком идентифицируют NP-трудность сформулированной подзадачи кластеризации.

Ее постановка не позволяет использовать какие-либо специальные методы решения, основанные на релаксации, например транспортной задачей, как в [31]. В данном случае применимы только общие методы решения задач псевдодулевого программирования, например, соответствующие встроенные средства систем

оптимизации IBM ILOG CPLEX Optimization studio, Gurobi, Xpress и др. На взгляд авторов, наибольшие перспективы имеет применение метода бинарных отсечений [37], однако объем статьи не позволяет изложить даже суть метода без привязки к задаче кластеризации. Поэтому ограничимся отсылкой к первоисточнику, ибо это тема отдельного исследования.

ЗК (13)–(17), напротив, имеет существенное множество алгоритмов решения. Одно перечисление таких алгоритмов, например [7–27], может занять несколько десятков страниц текста. Выделим только наилучшим образом зарекомендовавшие себя на практике алгоритмы, например, Кристофидеса [4], Лина–Кернигана [5, 6], а также работы последних лет, большая часть которых посвящена метаэвристическим методам: имитирующим физические процессы [8–12], методам «ровного» интеллекта [14–23] и эволюционным методам [13, 25].

### 3. Иллюстративный пример

Рассмотрим URPM-задачу с данными из [38] (табл. 1).

Таблица 1 / Table 1

Исходные данные / Initial data						
	1	2	3	4	5	6
1	$\infty$	27	43	16	30	26
2	7	$\infty$	16	1	30	30
3	20	13	$\infty$	35	5	0
4	21	16	25	$\infty$	18	18
5	12	46	27	48	$\infty$	5
6	23	5	5	9	5	$\infty$

Используя постановки (5)–(12) и (13)–(17), найдем решения для одного, двух и трех приборов равной производительности. Последнее означает равенство элементов  $c_{i,j}^k$  для всех  $k$ ,  $k = \overline{1,3}$ . Табл. 1 содержит эти элементы для шести заявок ( $n = 6$ ). Полагаем  $c_{j,j}^k = \infty$ .

1. Положим  $m = 1$ . Задача свелась к единственной ЗК в постановке (13)–(17). Ее решение представлено в табл. 2.

Таблица 2 / Table 2

$m = 1$						
U=	1	2	3	4	5	6
	0	4	1	0	2	3
1	0	0	0	1	0	0
2	1	0	0	0	0	0
3	0	0	0	0	1	0
4	0	0	1	0	0	0
5	0	0	0	0	0	1
6	0	1	0	0	0	0

Оценка быстродействия для данного случая  $\tilde{\Lambda} = 63$ .

2. Положим  $m = 2$ . Решаем подзадачу (5)–(11). Результаты сведены в табл. 3.

Назначения первого прибора – в левой части табл. 3, второго прибора – в ее правой части. Поскольку в решениях отсутствуют подциклы, полученные результаты являются оптимальным решением URPM-задачи с оценкой  $\tilde{\Lambda} = \max\{31, 37\} = 37$ .

Таблица 3 / Table 3

**$m = 2$**

	$k = 1$						$k = 2$					
	1	2	3	4	5	6	1	2	3	4	5	6
1	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	0	0	0

2. Положим  $m = 3$ . Решаем подзадачу (5)–(11). Результаты сведены в табл. 4.

Назначения первого прибора отмечены заливкой в левой части табл. 4, второго прибора – в средней части таблицы, назначения третьего прибора выделены заливкой ее правой части. Поскольку в решениях отсутствуют подциклы, полученные результаты являются оптимальным решением URPM-задачи с оценкой  $\tilde{\Lambda} = \max \{34, 27, 32\} = 34$ .

Таблица 4 / Table 4

**$m = 3$**

	$k = 1$						$k = 2$						$k = 3$					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

#### 4. Тестирование свойств программной реализации

Программная реализация предложенного подхода на данном этапе опирается на средства IBM ILOG CPLEX Optimization studio и включает формирование файлов данных подзадачи кластеризации (5)–(12) в соответствующих форматах, подключение модулей IBM ILOG CPLEX по технологии Concert, решение и сохранение результатов расчетов  $\tilde{x}_{i,j}^k, i, j \in \tilde{J}^k, k = \overline{1, m}, \lambda_{\min}(\tilde{x}_{i,j}^k)$  в формате csv. Далее последовательное формирование  $m$  ЗК (13)–(17), их решение и запись результатов  $\tilde{x}_{i,j}^k, i, j \in \tilde{J}^k, k = \overline{1, m}, \tilde{\Lambda} = \max \Lambda^k$  с предельной оценкой возможного отклонения от оптимума, абсолютной  $\Delta = \tilde{\Lambda} - \lambda_{\min}(\tilde{x}_{i,j}^k)$  и относительной

$$\delta = \frac{\Delta}{\lambda_{\min}(\tilde{x}_{i,j}^k)} \cdot 100\%.$$

Эмпирические оценки, наихудшие и средние, сформированы в результате генерации и расчетов тестовых примеров разной размерности. Результаты сведены в табл. 5. Сгенерировано около 100 примеров разных размерностей (размерности отображены в таблице).

Обозначения в таблицах соответствуют введенным выше:  $\lambda_{\min}$  – минимальное значение  $\lambda$  критерия (11) задачи кластеризации;  $t$  – общее время поиска решения URPM-теста в форматах сек. (секунды), либо чч:мм:сс (часы: минуты:

секунды) с оценкой  $\delta$ ;  $t_{avr}$  и  $\delta_{avr}$  – соответственно среднее время счета теста и средняя по группе тестов оценка максимально возможного отклонения от оптимума;  $n_x$  – число булевых переменных  $x_{i,j}^k$  в подзадаче кластеризации;  $m_r$  – общее число ограничений в подзадаче кластеризации.

Размерности тестов: число булевых переменных  $n_x = n^2 \cdot m$ . Минимальное и максимальное значения  $n_x = 30 \cdot 30 \cdot 2 = 1800$  и  $n_x = 100 \cdot 100 \cdot 4 = 40000$ ; число ограничений  $m_r = 2 \cdot n \cdot m + 2 \cdot n + m$ , минимальное и максимальное, соответственно  $m_r = 2 \cdot 30 \cdot 2 + 2 \cdot 30 + 2 = 182$  и  $m_r = 2 \cdot 100 \cdot 8 + 2 \cdot 100 + 8 = 1808$ .

Другими особенностями тестовых примеров, существенно влияющими на результаты и их оценки, являются параметры датчиков случайных чисел, генерирующие с равной вероятностью значения  $c_{i,j}^k$  в диапазоне чисел от 0 до 100. Теоретически это может приводить к фактическим отклонениям оптимальных решений  $\tilde{\Lambda}$  от  $\lambda$  (11) на десятки тысяч процентов. Для обеспечения сопоставимости результатов расчетов для одинаковых значений номера теста и числа заявок  $n$  использованы одинаковые значения  $c_{i,j}^k$ . Например, анализируя значения  $\tilde{\Lambda}$  для первого теста при  $n = 30$ ,  $m = 3$  и  $m = 4$ , для которых все  $c_{i,j}^k$  совпадают, непосредственно приходим к выводу о том, что решение по критерию быстродействия для трех приборов абсолютно лучше решения для четырех приборов (табл. 5).

Таблица 5 / Table 5

		n = 30					n = 50				
		№	1	2	3	4	5	1	2	3	4
m = 2	t	0	0	0	0	0	1	2	1	2	2
	$\lambda_{\min}$	78	102	71	86	74	79	73	82	72	63
	$\tilde{\Lambda}$	99	119	105	92	88	85	78	91	95	78
	$\delta$	26,92	16,67	47,89	6,98	18,92	7,59	6,85	10,98	31,94	23,81
m = 3	t	3	3	3	4	5	42	38	15	14	10
	$\lambda_{\min}$	54	70	51	59	54	55	49	56	49	42
	$\tilde{\Lambda}$	78	70	102	90	88	79	55	82	69	61
	$\delta$	44,44	0,00	100,00	52,54	62,96	43,64	12,24	46,43	40,82	45,24
m = 4	t	19	26	31	28	13	0:14:33	0:02:19	0:04:05	0:00:47	0:03:09
	$\lambda_{\min}$	42	56	41	46	42	43	38	43	38	34
	$\tilde{\Lambda}$	104	113	55	67	56	47	61	59	89	43
	$\delta$	147,62	101,79	34,15	45,65	33,33	9,30	60,53	37,21	134,21	26,47
m = 5	t	13	31	20	62	53	0:50:34	0:01:14	0:15:26	0:11:36	0:12:58
	$\lambda_{\min}$	36	47	35	38	37	35	30	36	32	28
	$\tilde{\Lambda}$	88	47	54	89	78	50	79	84	85	36
	$\delta$	144,44	0,00	54,29	134,21	110,81	42,86	163,33	133,33	165,63	28,57
m = 6	t	0:03:27	0:00:53	0:01:58	1:08:32	0:00:31		0:10:48			0:08:52
	$\lambda_{\min}$	33	46	31	35	33		27			24
	$\tilde{\Lambda}$	74	46	122	111	64		63			85
	$\delta$	124,24	0,00	293,55	217,14	93,94		133,33			254,17



Продолжение табл. 5

		$n = 70$					$n = 100$	
		№	1	2	3	4	5	1
$m = 2$	$t$		2	5	4	4	4	2
	$\lambda_{\min}$		58	58	58	65	64	65
	$\tilde{\lambda}$		64	62	60	72	76	75
	$\delta$		10,34	6,90	3,45	10,77	18,75	15,38
$m = 3$	$t$		0:01:19	0:01:09	0:01:13	0:00:50	0:00:32	0:01:35
	$\lambda_{\min}$		40	39	39	44	43	44
	$\tilde{\lambda}$		63	48	62	69	63	64
	$\delta$		57,50	23,08	58,97	56,82	46,51	45,45
$m = 4$	$t$		0:17:43	0:41:35	1:07:00	0:06:10	0:01:45	3:42:01
	$\lambda_{\min}$		31	30	31	34	33	34
	$\tilde{\lambda}$		50	49	44	53	51	57
	$\delta$		61,29	63,33	41,94	55,88	54,55	67,65
$m = 5$	$t$		0:42:34	2:59:07	0:52:30	8:32:51	0:06:28	
	$\lambda_{\min}$		25	25	25	28	27	
	$\tilde{\lambda}$		58	25	47	39	47	
	$\delta$		132,00	0,00	88,00	39,29	74,07	
$m = 6$	$t$						1:03:50	
	$\lambda_{\min}$						23	
	$\tilde{\lambda}$						47	
	$\delta$						104,35	

По меркам задач дискретного программирования сгенерированные тесты имеют значительные размерности. В силу отмеченной NP-полноты подзадачи кластеризации (5)–(11) с верхней оценкой числа вариантов решений  $2^{n_x}$  такая оценка трудоемкости тестовых задач определяется, казалось бы, фантастической величиной  $2^{40000}$ . Тем не менее программная реализация алгоритма поиска решений представленных выше задач относительно успешно с тестами справилась.

Как видно из табл. 5, предельное время счета теста составило более 8,5 часов для  $n = 70$  и  $m = 5$  ( $n_x = 24500$ ,  $m_r = 845$ , выделено заливкой в продолжении табл. 5).

Тесты, для которых решение за предельное время получить не удалось, также выделены заливкой в (незаполненных) графах времени счета.

Краткий анализ полученных результатов позволяет сделать следующие выводы. Наилучшие (ближайшие к оптимальным) решения удалось получить практически во всех тестах для двух приборов. В этих случаях средние значения относительной оценки отклонения  $\delta_{avr}$  лежат в интервале от 10 % (при  $n = 70$ ) до 23 % (при  $n = 30$ ). Среднее время счета находится в интервале от 5 до 0 секунд в той же последовательности. Тесты с тремя приборами определили интервал для  $\delta_{avr}$  от 38 % при  $n = 50$  до 52 % при  $n = 30$ . Среднее время счета составило для  $n = 30$  – 3,6 с, для  $n = 50$  – 23,8 с, для  $n = 70$  – 44 с, для  $n = 100$  – 1 мин 35 с. В тестах с четырьмя приборами среднее отклонение составило 53–55 % для  $n = 50$  и  $n = 70$  и 73 % для  $n = 30$ . При этом существенно увеличивается время счета от 23 с для  $n = 30$ , 5 мин для  $n = 50$ , 27 мин для  $n = 70$  и 3 ч 42 мин для  $n = 100$ . Решения тестовых задач с пятью приборами за установленный лимит времени удалось найти

только для значений  $n$  30, 50 и 70 при фатальном росте времени счета до предельного уровня в одном из тестов при  $n = 70$  (см. табл. 5). С увеличением числа приборов фиксируется также в среднем снижение оценок близости к оптимумам  $\delta_{avr}$ .

В целом полученные эмпирические оценки с учетом свойств тестовых задач можно признать вполне удовлетворительными. Проведенный анализ показал также целесообразность декомпозиции подзадачи кластеризации (5)–(11), что позволяет существенно снизить общее время счета без снижения оценки точности решения общей URPM-задачи.

Декомпозиция может быть осуществлена за счет разбиения множества приборов и последовательного решения модификаций условий (10) подзадачи (5)–(11) следующего вида:

$$\sum_{j=1}^n \sum_{i=1}^n c_{i,j}^k x_{i,j}^k \leq b^k \lambda, \quad k = \overline{1, m}, \quad (18)$$

где

$$b^k \geq 0, \quad k = \overline{1, m}, \quad \text{и} \quad \sum_{k=1}^m b^k = m. \quad (19)$$

Выражение (18) удобно также интерпретировать в форме  $\frac{1}{b^k} \sum_{j=1}^n \sum_{i=1}^n c_{i,j}^k x_{i,j}^k \leq \lambda$ .

Выражения (19) априори задают свойства весовых коэффициентов кластеров.

Последовательное решение подзадач (5)–(9), (18), (19), (11) позволяет исходную задачу кластеризации (5)–(11) решать приближенно в несколько этапов. Во-первых, это не ухудшает оценок для исходной URPM-задачи; во-вторых, радикально снижает проблему трудоемкости, выявленную на тестовых примерах. В частности, на каждом из шагов можно положить значение  $m = 2$ , что обеспечивает минимальное время счета, а за счет подбора последовательности весовых коэффициентов  $b^k \geq 0$ ,  $k = \overline{1, m}$ , – равномерность конечного разбиения на кластеры.

### Заключение

Анализ быстродействия программной реализации предложенного подхода, а также эмпирические оценки близости к оптимальным конструируемым расписаний выявили его высокую эффективность.

В целом разработанный инструментарий непосредственно применим для решения ряда важнейших практических задач с получением доказываемых оптимумов. Примером одного из подобных применений может служить решение задачи оптимизации календарных графиков бурения скважин (да и обустройства в целом) крупных нефтегазоконденсатных месторождений (НГКМ) [37]. Так, например, число кустов скважин одного из крупнейших НГКМ Западной Сибири составляет 60–70 (количество заявок), число используемых кустовых буровых установок 7–10 (количество приборов). Рассмотренная нами URPM-проблема и средства ее решения в данном контексте (с учетом размерности) позволяют найти точные, либо близкие к оптимальным по быстродействию (с оценками возможного отклонения) календарные графики бурения с определением маршрутов перемещений буровых, временных и ресурсных характеристик. По самым скромным оценкам это дает за счет сжатия календарных графиков не менее 10 % экономии при достижении плановых объемов меньшим числом буровых за плановый период, либо 10 % увеличения добычи за плановый период без увеличения числа буровых.

## ЛИТЕРАТУРА

1. An algorithm for the traveling salesman problem / J.D.C. Little, K.G. Murty, D.W. Sweeney, C. Karel // *Operations Research*. – 1963. – Vol. 11. – P. 972–989.
2. **Беллман Р.** Применение динамического программирования к задаче о коммивояжере // *Кибернетический сборник*. – 1964. – № 9. – С. 219–222.
3. **Miliotis P.** Integer programming approaches to the travelling salesman problem // *Mathematical Programming*. – 1976. – Vol. 10, iss. 3. – P. 367–378.
4. **Christofides N.** Worst-case analysis of a new heuristic for the traveling salesman problem // *Symposium on New Directions and Recent Results in Algorithms and Complexity*. – Orlando: Academic Press, 1976. – P. 441.
5. **Lin S., Kernighan B.W.** An effective heuristic algorithm for the traveling salesman problem // *Operations Research*. – 1973. – Vol. 21, iss. 2. – P. 498–516.
6. **Helsgaun K.** An effective implementation of the Lin-Kernighan traveling salesman heuristic // *European Journal of Operational Research*. – 2000. – Vol. 126, iss. 1. – P. 106–130.
7. **Golden B.L., Skiscim C.C.** Using simulated annealing to solve routing and location problems // *Naval Research Logistics Quarterly*. – 1986. – Vol. 33, iss. 2. – P. 261–279.
8. **Haist T., Osten W.** An optical solution for the traveling salesman problem // *Optics Express*. – 2007. – Vol. 15, iss. 16. – P. 10473–10482. – doi: 10.1364/OE.15.010473.
9. **Javadian N., Alikhani M.G., Tavakkoli-Moghaddam R.** A discrete binary version of the electromagnetism-like heuristic for solving traveling salesman problem // *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. – Berlin; Heidelberg: Springer, 2008. – P. 123–130. – doi: 10.1007/978-3-540-85984-0\_16.
10. **Bouzidi M., Riffi M.E.** Adaptation of the harmony search algorithm to solve the travelling salesman problem // *Journal of Theoretical & Applied Information Technology*. – 2014. – Vol. 62, N 1. – P. 154–160.
11. **Balachandar S.R., Kannan K.** Randomized gravitational emulation search algorithm for symmetric traveling salesman problem // *Applied Mathematics and Computation*. – 2007. – Vol. 192, iss. 2. – P. 413–421. – doi: 10.1016/j.amc.2007.03.019.
12. **Fiechter C.N.** A parallel tabu search algorithm for large traveling salesman problems // *Discrete Applied Mathematics*. – 1994. – Vol. 51, iss. 3. – P. 243–267. – doi: 10.1016/0166-218X(92)00033-I.
13. **Verma O.P., Jain R., Chhabra V.** Solution of travelling salesman problem using bacterial foraging optimisation algorithm // *International Journal of Swarm Intelligence*. – 2014. – Vol. 1, N 2. – P. 179–192. – doi: 10.1504/IJSI.2014.060243.
14. **Dorigo M., Gambardella L.M.** Ant colonies for the travelling salesman problem // *BioSystems*. – 1997. – Vol. 43, iss. 2. – P. 73–81. – doi: 10.1016/S0303-2647(97)01708-5.
15. **Wong L.P., Low M.Y.H., Chong C.S.** A bee colony optimization algorithm for traveling salesman problem // *Second Asia International Conference on Modelling & Simulation*, Kuala Lumpur, Malaysia, 13–15 May 2008. – Piscataway: IEEE, 2008. – P. 818–823. – doi: 10.1109/AMS.2008.27.
16. **Jati G.K.** Evolutionary discrete firefly algorithm for travelling salesman problem. – Berlin; Heidelberg: Springer, 2011. – P. 393–403. – doi: 10.1007/978-3-642-23857-4\_38.
17. **Feng X., Lau F.C.M., Yu H.** A novel bio-inspired approach based on the behavior of mosquitoes // *Information Sciences*. – 2013. – Vol. 233. – P. 87–108. – doi: 10.1016/j.ins.2012.12.053.
18. **Xue-Hui L., Ye Y., Xia L.** Solving TSP with shuffled frog-leaping algorithm // *Eighth International Conference on Intelligent Systems Design and Applications, ISDA 2008*. – Los Alamitos, 2008. – Vol. 3. – P. 228–232. – doi: 10.1109/ISDA.2008.346.
19. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems / E. Osaba, X.-S. Yang, F. Diaz, П. Lopez-Garcia, R. Carballo // *Engineering Applications of Artificial Intelligence*. – 2016. – Vol. 48. – P. 59–71. – doi: 10.1016/j.engappai.2015.10.006.
20. **Mzili I., Riffi M.E.** Discrete penguins search optimization algorithm to solve the traveling salesman problem // *Journal of Theoretical & Applied Information Technology*. – 2015. – Vol. 72, N 3. – P. 331–336.
21. **Bouzidi A., Riffi M.E.** Discrete cat swarm optimization to resolve the traveling salesman problem // *International Journal of Advanced Research in Computer Science and Software Engineering*. – 2013. – Vol. 3, iss. 9. – P. 13–18. – doi: 10.1109/WCCCS.2014.7107914.

22. **Agharghor A., Riffi M.E.** Hunting search algorithm to solve the traveling salesman problem // *Journal of Theoretical & Applied Information Technology*. – 2015. – Vol. 74, N 1. – P. 120–129.
23. Producer-scrounger method to solve traveling salesman problem / M.A.H. Akhand, Pintu Chnadra Shill, Md. Forhad Hossain, A.B.M. Junaed, K. Murase // *International Journal of Intelligent Systems and Applications*. – 2015. – Vol. 7, N 3. – P. 29–36. – doi: 10.5815/ijisa.2015.03.04.
24. **Goldbarg E.F.G., Souza G.R. de, Goldbarg M.C.** Particle swarm optimization algorithm for the traveling salesman problem [Electronic resource] // *Traveling Salesman Problem*. – 2008. – URL.: [http://www.intechopen.com/books/traveling\\_salesman\\_problem/particle\\_swarm\\_optimization\\_algorithm\\_for\\_the\\_traveling\\_salesman\\_problem](http://www.intechopen.com/books/traveling_salesman_problem/particle_swarm_optimization_algorithm_for_the_traveling_salesman_problem) (accessed: 07.04.2016).
25. **Potvin J.Y.** Genetic algorithms for the traveling salesman problem // *Annals of Operations Research*. – 1996. – Vol. 63, iss. 3. – P. 337–370. – doi: 10.1007/BF02125403.
26. **Ouaarab A., Ahiod B., Yang X.S.** Discrete cuckoo search algorithm for the travelling salesman problem // *Neural Computing and Applications*. – 2014. – Vol. 24, iss. 7/8. – P. 1659–1669. – doi: 10.1007/s00521-013-1402-2.
27. **Clarke G., Wright J.W.** Scheduling of vehicles from a central depot to a number of delivery points // *Operations Research*. – 1964. – Vol. 12, N 4. – P. 568–581.
28. **Костюк Ю.Л., Пожидаев М.С.** Приближенные алгоритмы решения сбалансированной задачи к коммивояжеров // *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*. – 2008. – № 1. – С. 106–112.
29. Solution of a min-max vehicle routing problem / D. Applegate, W. Cook, S. Dash, A. Rohe // *INFORMS Journal on Computing*. – 2002. – Vol. 14, iss. 2. – P. 132–143. – doi: 10.1287/ijoc.14.2.132.118.
30. **Ren C.** Solving min-max vehicle routing problem // *Journal of Software*. – 2011. – Vol. 6, N 9. – P. 1851–1856. – doi: 10.4304/jsw.6.9.1851-1856.
31. **Алексеев А.О.** Минимаксная задача М коммивояжеров // *Журнал вычислительной математики и математической физики*. – 1991. – Т. 31, № 12. – С. 1899–1905.
32. **Svestka J.A., Huckfeldt V.E.** Computational experience with an m-salesman traveling salesman algorithm // *Management Science*. – 1973. – Vol. 19, iss. 7. – P. 790–799.
33. **Bektas T.** The multiple traveling salesman problem: an overview of formulations and solution procedures // *Omega*. – 2006. – Vol. 34, iss. 3. – P. 209–219. – doi: 10.1016/j.omega.2004.10.004.
34. **Мезенцев Ю.А.** Оптимизация расписаний параллельных динамических систем в календарном планировании // *Информационные технологии*. – 2008. – № 2. – С. 16–23.
35. **Мезенцев Ю.А.** Оптимизация расписаний параллельно-последовательных систем в календарном планировании // *Информационные технологии*. – 2009. – № 6. – С. 35–41.
36. **Мезенцев Ю.А.** Эффективные вычислительные методы решения дискретных задач оптимизации управления производственными процессами. – Новосибирск: Изд-во НГТУ, 2015. – 275 с.
37. **Мезенцев Ю.А.** Метод бинарных отсечений и ветвлений целочисленного программирования // *Доклады Академии наук высшей школы Российской Федерации*. – 2011. – № 1 (16). – С. 12–25.
38. **Филлипс Д., Гарсиа-Диас А.** Методы анализа сетей. – М.: Мир, 1984. – 496 с.

## PROBLEMS AND OPTIMIZATION ALGORITHMS OF PARALLEL-SERIAL SYSTEMS SCHEDULES WITH UNDEFINED SERVICE ROUTES

**Mezentsev Yu.A., Estraykh I.V.**

*Novosibirsk State Technical University, Novosibirsk, Russia*

A new approach to solving problems of constructing optimal operating speed schedules for parallel-serial systems using a two-staged scheme including clustering and subsequent route determination is presented. Solutions to different types of routing problems with the minimax criterion are special cases of applying the developed tools. Formal definitions of the clustering and route design subproblems in the form of NP-difficult discrete programming problems are given. An approximate algorithm of problem solving is proposed. It is based on the problem decomposi-

tion into a subproblem optimal for partitioning a set of requests into subsets, with a load on the devices being uniform, and into a number of subproblems for optimizing scheduling jobs on the devices. Illustrative examples of applying the proposed approach are given. Empirical estimates of the accuracy and software performance are presented. The speed and accuracy of the developed algorithms are studied on randomly generated test examples in a wide range of dimensions. Possible practical applications of the proposed algorithms for scheduling the implementation of major industrial projects are described and further directions of the proposed approach development are identified.

*Keywords:* parallel-serial service system; undefined service routes; minimizing makespan; routing; clustering; traveling salesman problem.

DOI: 10.17212/1727-2769-2016-3-83-97

#### REFERENCES

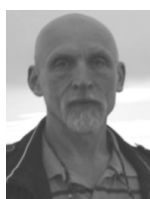
1. Little J.D.C., Murty K.G., Sweeney D.W., Karel C. An algorithm for the traveling salesman problem. *Operations Research*, 1963, vol. 11, pp. 972–989.
2. Bellman R. Primenenie dinamicheskogo programmirovaniya k zadache o kommivoyazhere [Dynamic programming treatment of the travelling salesman problem]. *Kiberneticheskii sbornik – Cybernetic digest*, 1964, no. 9, pp. 219–222.
3. Miliotis P. Integer programming approaches to the travelling salesman problem. *Mathematical Programming*, 1976, vol. 10, iss. 3, pp. 367–378.
4. Christofides N. Worst-case analysis of a new heuristic for the traveling salesman problem. *Symposium on New Directions and Recent Results in Algorithms and Complexity*. Orlando, Academic Press, 1976, p. 441.
5. Lin S., Kernighan B.W. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 1973, vol. 21, iss. 2, p. 498–516.
6. Helsgaun K. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 2000, vol. 126, iss. 1, pp. 106–130.
7. Golden B.L., Skiscim C.C. Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly*, 1986, vol. 33, iss. 2, pp. 261–279.
8. Haist T., Osten W. An optical solution for the traveling salesman problem. *Optics Express*, 2007, vol. 15, iss. 16, pp. 10473–10482. doi: 10.1364/OE.15.010473
9. Javadian N., Alikhani M.G., Tavakkoli-Moghaddam R. A discrete binary version of the electromagnetism-like heuristic for solving traveling salesman problem. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. Berlin, Heidelberg, Springer, 2008, pp. 123–130. doi: 10.1007/978-3-540-85984-0\_16
10. Bouzidi M., Riffi M.E. Adaptation of the harmony search algorithm to solve the travelling salesman problem. *Journal of Theoretical & Applied Information Technology*, 2014, vol. 62, no. 1, pp. 154–160.
11. Balachandar S.R., Kannan K. Randomized gravitational emulation search algorithm for symmetric traveling salesman problem. *Applied Mathematics and Computation*, 2007, vol. 192, iss. 2, pp. 413–421. doi: 10.1016/j.amc.2007.03.019
12. Fiechter C.N. A parallel tabu search algorithm for large traveling salesman problems. *Discrete Applied Mathematics*, 1994, vol. 51, iss. 3, pp. 243–267. doi: 10.1016/0166-218X(92)00033-I
13. Verma O.P., Jain R., Chhabra V. Solution of travelling salesman problem using bacterial foraging optimisation algorithm. *International Journal of Swarm Intelligence*, 2014, vol. 1, no. 2, pp. 179–192. doi: 10.1504/IJSI.2014.060243
14. Dorigo M., Gambardella L.M. Ant colonies for the travelling salesman problem. *BioSystems*, 1997, vol. 43, iss. 2, pp. 73–81. doi: 10.1016/S0303-2647(97)01708-5
15. Wong L.P., Low M.Y.H., Chong C.S. A bee colony optimization algorithm for traveling salesman problem. *Second Asia International Conference on Modelling & Simulation*, Kuala Lumpur, Malaysia, 13–15 May 2008, pp. 818–823. doi: 10.1109/AMS.2008.27
16. Jati G.K. *Evolutionary discrete firefly algorithm for travelling salesman problem*. Berlin, Heidelberg, Springer, 2011, pp. 393–403. doi: 10.1007/978-3-642-23857-4\_38.
17. Feng X., Lau F.C.M., Yu H. A novel bio-inspired approach based on the behavior of mosquitoes. *Information Sciences*, 2013, vol. 233, pp. 87–108. doi: 10.1016/j.ins.2012.12.053

18. Xue-Hui L., Ye Y., Xia L. Solving TSP with shuffled frog-leaping algorithm. *Eighth International Conference on Intelligent Systems Design and Applications, ISDA 2008*, vol. 3, pp. 228–232. doi: 10.1109/ISDA.2008.346
19. Osaba E., Yang X.-S., Diaz F., Lopez-Garcia P., Carballedo R. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence*, 2016, vol. 48, pp. 59–71. doi: 10.1016/j.engappai.2015.10.006
20. Mzili I., Riffi M.E. Discrete penguins search optimization algorithm to solve the traveling salesman problem. *Journal of Theoretical & Applied Information Technology*, 2015, vol. 72, no. 3, pp. 331–336.
21. Bouzidi A., Riffi M.E. Discrete cat swarm optimization to resolve the traveling salesman problem. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013, vol. 3, iss. 9, pp. 13–18. doi: 10.1109/WCCCS.2014.7107914
22. Agharghor A., Riffi M.E. Hunting search algorithm to solve the traveling salesman problem. *Journal of Theoretical & Applied Information Technology*, 2015, vol. 74, no. 1, pp. 120–129.
23. Akhand M.A.H., Pintu Chnadra Shill, Forhad Hossain Md., Junaed A.B.M., Murase K. Producer-scrounger method to solve traveling salesman problem. *International Journal of Intelligent Systems and Applications*, 2015, vol. 7, no. 3, pp. 29–36. doi: 10.5815/ijisa.2015.03.04
24. Goldberg E.F.G., Souza G.R. de, Goldberg M.C. Particle swarm optimization algorithm for the traveling salesman problem. *Traveling Salesman Problem*, 2008. Available at: [http://www.intechopen.com/books/traveling\\_salesman\\_problem/particle\\_swarm\\_optimization\\_algorithm\\_for\\_the\\_traveling\\_salesman\\_problem](http://www.intechopen.com/books/traveling_salesman_problem/particle_swarm_optimization_algorithm_for_the_traveling_salesman_problem) (accessed 07.04.2016)
25. Potvin J.Y. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, 1996, vol. 63, iss. 3, pp. 337–370. doi: 10.1007/BF02125403
26. Ouaarab A., Ahiod B., Yang X.S. Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Applications*, 2014, vol. 24, iss. 7–8, pp. 1659–1669. doi: 10.1007/s00521-013-1402-2
27. Clarke G., Wright J.W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 1964, vol. 12, no. 4, pp. 568–581.
28. Kostyuk Yu.L., Pozhidaev M.S. Priblizhennyye algoritmy resheniya sbalansirovannoi zadachi k kommivoyazherov [Approximate algorithms for solution the balanced problem of k travel salesman]. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*, 2008, no. 1, pp. 106–112.
29. Applegate D., Cook W., Dash S., Rohe A. Solution of a min-max vehicle routing problem. *INFORMS Journal on Computing*, 2002, vol. 14, iss. 2, pp. 132–143. doi: 10.1287/ijoc.14.2.132.118
30. Ren C. Solving min-max vehicle routing problem. *Journal of Software*, 2011, vol. 6, no. 9, pp. 1851–1856. doi: 10.4304/jsw.6.9.1851–1856
31. Alekseev A.O. Minimaksnaya zadacha M kommivoyazherov [A minimax problem of M traveling salesmen]. *Zhurnal vychislitel'noi matematiki i matematicheskoi fiziki – Computational Mathematics and Mathematical Physics*, 1991, vol. 31, no. 12, pp. 1899–1905. (In Russian)
32. Svestka J.A., Huckfeldt V.E. Computational experience with an m-salesman traveling salesman algorithm. *Management Science*, 1973, vol. 19, iss. 7, pp. 790–799.
33. Bektas T. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 2006, vol. 34, iss. 3, pp. 209–219. doi: 10.1016/j.omega.2004.10.004
34. Mezentsev Yu.A. Optimizatsiya raspisanii parallel'nykh dinamicheskikh sistem v kalendar-nom planirovanii [Optimization of schedules for dynamic parallel service systems]. *Informatsionnye tekhnologii – Information Technology*, 2008, no. 2, pp. 16–23.
35. Mezentsev Yu.A. Optimizatsiya raspisanii parallel'no-posledovatel'nykh sistem v kalendar-nom planirovanii [Optimization of schedules in parallel-serial systems for scheduling]. *Informatsionnye tekhnologii – Information Technology*, 2009, no. 6, pp. 35–41.
36. Mezentsev Yu.A. *Effektivnye vychislitel'nye metody resheniya diskretnykh zadach optimizatsii upravleniya proizvodstvennymi protsessami* [Effective numerical methods for solution of discrete optimization problems management]. Novosibirsk, NSTU Publ., 2015. 275 p.
37. Mezentsev Yu.A. Metod binarnykh otsechenii i vetvlenii tselochislennogo programirovaniya [Branch and binary cuts approach of integer programming]. *Doklady Akademii*

*nauk vysshei shkoly Rossiiskoi Federatsii – Proceedings of the Russian higher school Academy of sciences*, 2011, no. 1 (16), pp. 12–25.

38. Phillips D., Garcia-Diaz A. *Fundamental of network analysis*. Englewood Cliffs, Prentice-Hall, 1981 (Russ. ed.: Fillips D., Garsia-Dias A. *Metody analiza setei*. Moscow, Mir Publ., 1984. 496 p.).

## СВЕДЕНИЯ ОБ АВТОРАХ



**Мезенцев Юрий Анатольевич** – родился в 1955 году, д-р техн. наук, доцент, профессор кафедры экономической информатики Новосибирского государственного технического университета. Область научных интересов: задачи и методы дискретного программирования. Опубликовано 55 научных работ. (Адрес: 630073, Россия, Новосибирск, проспект Карла Маркса 20. Email: mezencev@corp.nstu.ru).

**Mezentsev Yuriy Anatoliyovich** (b. 1955) – Doctor of Sciences (Eng.), Professor at the Computer Science in Economics Department in the Novosibirsk State Technical University. His research interests are currently focused on problems and methods of discrete programming. He is author of 55 scientific papers. (Address: 20, Karl Marx Av., Novosibirsk, 630073, Russia. Email: mezencev@corp.nstu.ru).



**Эстрайх Игорь Викторович** – родился в 1961 году, старший преподаватель кафедры экономической информатики Новосибирского государственного технического университета. Область научных интересов: задачи и методы дискретного программирования. (Адрес: 630073, Россия, Новосибирск, проспект Карла Маркса 20. Email: estrajx@corp.nstu.ru).

**Estraykh Igor Viktorovich** (b. 1961) – Senior Lecturer at the Computer Science in Economics Department in the Novosibirsk State Technical University. His research interests are currently focused on problems and methods of discrete programming. (Address: Address: 20, Karl Marx Av., Novosibirsk, 630073, Russia. Email: estrajx@corp.nstu.ru).

*Статья поступила 04 марта 2016 г.  
Received March 04, 2016*

## To Reference:

Mezentsev Y.A., Estraykh I.V. Zadachi i algoritmy optimizatsii raspisanii parallelnoposledovatel'nykh sistem s neopredelennymi marshrutami obsluzhivaniya [Problems and optimization algorithms of parallel-serial systems schedules with undefined service routes]. *Doklady Akademii nauk vysshei shkoly Rossiiskoi Federatsii – Proceedings of the Russian higher school Academy of sciences*, 2016, no. 1 (30), pp. 83–97. doi: 10.17212/1727-2769-2016-3-83-97